
Der Schutz von Computersoftware – Urheberrecht oder Sonderrecht?

Dissertation der Rechts- und staatswissenschaftlichen
Fakultät der Universität Zürich
zur Erlangung der Würde eines Doktors
der Rechtswissenschaft,

vorgelegt von

Hans Rudolf Wittmer

von Niedererlinsbach SO

Genehmigt auf Antrag von Herrn
Professor Dr. Manfred Rehbinder

1981



STÄMPFLI & CIE AG BERN



Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41507203_0066

Der Schutz von Computersoftware – Urheberrecht oder Sonderrecht?

Dissertation der Rechts- und staatswissenschaftlichen
Fakultät der Universität Zürich
zur Erlangung der Würde eines Doktors
der Rechtswissenschaft,

vorgelegt von

Hans Rudolf Wittmer

von Niedererlinsbach SO

Genehmigt auf Antrag von Herrn
Professor Dr. Manfred Rehbinder

1981



STÄMPFLI & CIE AG BERN

Die Rechts- und staatswissenschaftliche Fakultät gestattet hierdurch die Drucklegung vorliegender Dissertation, ohne damit zu den darin ausgesprochenen Anschauungen Stellung zu nehmen.

Zürich, den 9. Juli 1980

Der Dekan:
Prof. Dr. G. Hauser

Die gleiche Arbeit ist im Verlag Stämpfli & Cie AG Bern erschienen als Heft 6 der Reihe

Schriften zum Medienrecht SM

begründet und herausgegeben von Prof. Dr. Manfred Rehbinden und PD. Dr. Wolfgang Larese.



©

Verlag Stämpfli & Cie AG Bern 1981
Rechnergesteuerter Filmsatz, Offsetdruck und
Buchbinderarbeit Stämpfli & Cie AG Bern
Printed in Switzerland

Inhaltsverzeichnis

Abkürzungsverzeichnis	11
Literaturverzeichnis	12
Quellenverzeichnis	18
1. Kapitel: Einführung in die Problematik und Vorgehen	19
A. Relevanz und Vorgehen	19
B. Einführung in den Stand der Diskussion	19
C. Vorläufige Lokalisierung der Problematik	22
2. Kapitel: Grundlagen für die Charakterisierung der infragestehenden Schutzobjekte	25
A. Allgemeine Begriffe der Elektronischen Datenverarbeitung	25
B. Was ist Software?	26
I. Grundlagen	26
1. Das Verhältnis von Software und Hardware	26
2. Elemente und Grundtypen von Programmen	27
3. Die softwaremässige Kompatibilität	29
4. Das Programm als Bestandteil der Software	30
II. Der dieser Arbeit zugrundegelegte Softwarebegriff	31
Zusammenfassung	35
C. Phasen der Software-Entwicklung (Programmierungsphasen)	35
I. Gegenstand und Verhältnis der Entwicklungsphasen untereinander	35
1. Sphäre der Aufgaben- oder Problemstellung	36
2. Systemsphäre	36
3. Programmsphäre	36
II. Die urheberrechtliche Relevanz der Entwicklungs- oder Programmierungsphasen ..	37
1. Die nach Programmierungsphasen differenzierende Betrachtung	37
2. Die Abwendung von einer nach Programmierungsphasen differenzierenden Be- trachtung	38
a) Tatsächliches	38
b) Rechtliches	40
Ergebnis	42
D. Der Erkenntnisgegenstand von Algorithmus und Programm	42
I. Bedingungen eines Algorithmus	43
II. Anwendung dieser Bedingungen auf das Programm	43
III. Zusammenfassung	43

E.	Katalog der wichtigsten Ergebnisse der Softwareherstellung	44
I.	Umgangssprachlich geformte Ergebnisse, Software-Dokumentationen	44
	1. Allgemeine Systembeschreibung	44
	2. (Umgangssprachliche) Programmbeschreibung	45
	3. Datei- und Datenbeschreibungen	45
	4. Bedienungsanleitungen, Anwendungshandbücher (Manuals)	45
	5. Weitere Ergebnisse dieser Gruppe	45
II.	Ergebnisse mit zeichnerischen, schematischen oder tabellarischen Darstellungen ...	45
	1. Flussdiagramme verschiedener Ebenen	46
	2. Layouts von Ein- und Ausgabemedien	46
	3. Weitere Ergebnisse dieser Gruppe	47
III.	Sprachlich geformte Ergebnisse mit Verwendung einer genormten maschinen-, sym- bol-, problem- oder generator-orientierten Kunstsprache	47
	1. Maschinensprachen und maschinenorientierte Assemblersprachen	48
	2. Höhere symbolische Programmiersprachen	49
	3. Norm- und Struktursprachen	49
	4. Generatorsprachen	49
	5. Andere Sprachen	50
	6. Programm-Standards, Programmrichtlinien	50
	7. Source- und Objektsprachen und deren Formen	50
	8. Ausblick	51
IV.	Softwarebestandteile mit üblicherweise nur mittelbar wahrnehmbarer Verkörperung auf magnetischen Datenträgern	51
	1. Referenz-, Parameter-, Textdateien und sonstige Dateien mit Steuerungsinformatio- nen	52
	2. Testdaten	52
3. Kapitel: Beurteilung der rechtlichen Schutzerfordernisse aufgrund der Schutzinteressen der Beteiligten und der ausserrechtlichen Schutz- möglichkeiten		53
A.	Entwicklungen in der Softwareherstellung und -vermarktung	53
	Zusammenfassung	56
B.	Der «Selbstschutz» und Massnahmen zum Schutz der Software	57
I.	Der «Selbstschutz» der Software	58
	1. Eignungsprüfung und Realisierung des praktischen Einsatzes	58
	2. «Selbstschutz» wegen der Fehlerhaftigkeit und der Notwendigkeit dauernder Pro- grammwartung	59
II.	Organisatorisch-administrative und softwaremässige Massnahmen zum Schutz der Software (Datensicherung)	60
III.	Vorsichtsmassnahmen bei der Verbreitung der Software	62
	Ergebnis	63
C.	Die Beurteilung der mit dem Softwareschutz verknüpften Interessen	63
I.	Zum Interessenverhältnis von Hardware- und reinen Softwareherstellern (Software- häuser)	64
II.	Die Interessen der Allgemeinheit	65
	Zusammenfassung	69

4. Kapitel: Rechtliche Schutzmöglichkeiten der Software ausserhalb des Urheberrechtes und eines Sonderschutzrechtes de lege ferenda	70
A. Einleitung	70
B. Der vertragsrechtliche Schutz	70
C. Der wettbewerbsrechtliche Schutz	71
I. Grenzen des Geheimnisschutzes	72
II. Das Wettbewerbsverhältnis	73
III. Unsicherheiten bei der Anwendung der Generalklausel des Art. 1 Abs. 1 UWG	75
1. Im besonderen zur sklavischen Nachahmung	75
2. Im besonderen zur unmittelbaren Leistungsübernahme	76
Ergebnis	78
D. Der patentrechtliche Schutz	78
Ergebnis	81
5. Kapitel: Urheberrechtliche Grundlagen	82
A. Zum Vorgehen	82
B. Die Grundlagen zu den umstrittenen urheberrechtlichen Problemfelder	83
I. Die Zuordnung zu bestimmten Werkkategorien	84
1. Die Zuordnung der umgangssprachlichen Dokumentationen und kunstsprachlichen Programme	84
2. Die Zuordnung von zeichnerischen, schematischen und tabellarischen Darstellungen mit sprachlichen Elementen	86
II. Der urheberrechtliche Schutz von wissenschaftlichen Werken	86
Vorläufige Übersicht und Zusammenfassung	89
1. Die Individualität der wissenschaftlichen Erkenntnisse (Inhalt)	90
2. Die Zweckfreiheit	92
3. Die Anweisungen	94
4. Individuelle Elemente in- und ausserhalb der schützbaren Substanz	99
5. Das Problem der Mitteilungsexklusivität	101
6. Die Form	103
a) Das Ausdrucksmittel	103
b) Die äussere und die innere Form	103
Ergebnis	104
6. Kapitel: Die urheberrechtliche Beurteilung der einzelnen Softwarebestandteile	106
A. Softwareergebnisse mit vorwiegend umgangssprachlichen Elementen	106
I. Die Form	106
II. Der Inhalt	107
B. Softwareergebnisse mit grafischen oder schematischen Darstellungen	107
I. Flussdiagramme verschiedener Ebenen	107
1. Die Form	107

	a) Zeichnerische, darstellerische Elemente	108
	b) Sprachliche Elemente	108
2.	Der Inhalt	109
II.	Layouts von Ein- und Ausgabemedien, insbesondere Druck- und Bildschirmbilder ..	110
1.	Abgrenzung zwischen Ein- und Ausgabedaten und Elementen der Software	110
2.	Die Form der Layouts	112
	a) Bildschirmlayouts	112
	b) Drucklayouts	113
	c) Datensatz- (Record-) Layouts	113
3.	Der Inhalt der Layouts	113
C.	Softwareergebnisse in einer maschinell verwertbaren (kunst-) sprachlichen Form	114
I.	Individualitäten im Bereich der Form oder im «Wortlaut» der Programme	114
1.	Merkmalsbildung, Operandensymbolik, Schlüssel- oder Kennworte	116
2.	Die Kombination und Reihenfolge der Symbolik	116
3.	Programmerläuterungen, Kommentare	117
4.	Konstantendefinitionen und sonstige Bestandteile für wiederkehrende Ein- und Ausgaben (Bedienungsaufrufe und Elemente des Beiwerks von Bildschirm- und Druckbildern)	117
5.	Programmstruktur, innere Anordnung	118
6.	Programmierungskonventionen, Normierungen, Richtlinien	119
	Ergebnis	120
II.	Individualitäten im Bereich des Inhaltes von Programmen und von Softwarelösungen im allgemeinen	120
1.	Kenntnisse über das Vorhandensein von Grössen, die mit Hilfe eines Automaten umgeformt werden können	121
2.	Kenntnisse über das Vorhandensein und die Eigenschaften von Grössen, die mit Hilfe eines Automaten aus bestimmten Ausgangsgrössen gewonnen werden können	122
3.	Kenntnisse über das Vorhandensein von Umformungsprozessen (Operationen) für die Umformung von Eingangsgrössen in Ausgangsgrössen	122
4.	Kenntnisse über die logische Ordnung und die Reihenfolge des Ablaufes von automatisierten Umformungsprozessen	123
	a) Der Ablauf von verschiedenen Programmen, der Applikationsablauf	124
	b) Der Programmablauf	124
5.	Kenntnisse über Methoden zur Anpassung der erforderlichen Umformungsprozesse an die Eigenschaften eines Automaten	125
	Zusammenfassung	127
III.	Der Schutz von Programmen in verschiedenen Sprachen	128
1.	Maschinenorientierte oder maschinennahe Sprachen, einfache Symbolsprachen, Assemblersprachen u. dergl.	128
2.	Struktur- und Generatorsprachsysteme	129
IV.	Der Schutz von verschiedenen Programmarten	130
1.	Unterprogramme, Programmteile	130
2.	Objektprogramme oder maschinell verwertbare Zwischenformen	131
D.	Softwarebestandteile mit üblicherweise nur mittelbar wahrnehmbarer Verkörperung auf magnetischen Datenträgern	133
I.	Referenz-, Parameter-, Textdateien und sonstige Dateien mit programmergänzenden oder -ersetzenden Funktionen	133
II.	Testdateien	134

E.	Schutzzumfang und Schutzwirkung	134
I.	Einleitung	134
II.	Formschutz	137
III.	Einbezug des Schutzes von nicht gemeinfreien inhaltlichen Elementen	139
IV.	Nebenwirkungen des Urheberrechtes auf Programmier- und Betriebssystemsprachen, Betriebssystemtechniken und Interpretationsprogramme	140
V.	Zur gerichtlichen Durchsetzung des urheberrechtlichen Softwareschutzes	143
1.	Nachweis der Werkeigenschaft	143
2.	Beweis der Verletzungstatbestände	143
F.	Verschiedene Fragenkreise zum Urheberrecht der Software	145
I.	Die rechtliche Erfassung der Verwendungsformen von Softwarewerken	146
II.	Die Rechtszuordnung	150
III.	Die Schutzdauer	151
IV.	Die Neuheit	152
V.	Formalitäten	152
	Ergebnis	153
7. Kapitel:	Sonderschutz für Computersoftware de lege ferenda	155
A.	Einleitung	155
B.	Gegenüberstellung der urheberrechtlichen Ergebnisse mit den WIPO-Mustervorschriften	157
I.	§ 1 MV: Begriffsbestimmungen	157
II.	§ 1 (v) und § 2 MV: Rechtsinhaber	160
III.	§ 3 MV: Originalität	162
IV.	§ 4 MV: Konzepte	163
V.	§ 5 MV: Rechte des Inhabers	164
1.	§ 5 i, ii MV: Geheimhaltung	165
2.	§ 5 iii MV: Vervielfältigung	165
3.	§ 5 iv und v MV: Unveränderte nachschaffende Übernahme oder Nachahmung von Programmen und Programmbeschreibungen sowie Herstellung von gleichartigen Programmen nach Programmbeschreibungen	165
4.	§ 5 vi MV: Gebrauch und Einspeicherung	166
5.	§ 5 vii MV: Verschiedene Verbreitungszwecke	166
VI.	§ 6 MV: Rechtsverletzungen	166
VII.	§ 7 MV: Schutzdauer	167
VIII.	§ 8 MV: Rechtsbehelfe	167

IX.	§ 9 MV: Anwendbarkeit anderer Rechtsvorschriften	168
	Ergebnis	168

Anhang

Nr. 1	Die wichtigste Software für Datenverarbeitungsanlagen	170
Nr. 2	Logischer Ablauf bei der Erstellung eines Programmes	171
Nr. 3	System-/Datenflussdiagramm	172
Nr. 4	Programmablaufplan	173
Nr. 5	Bildschirmbilder/Bildschirmlayout	174
Nr. 6	Druckbilder/Drucklayout	175
Nr. 7	Programmbeispiel (Source-Programm)	176

Abkürzungsverzeichnis

ACM	Association for Computing Machinery (USA)
AIPPI	Association internationale pour la protection de la propriété intellectuelle
a. M./gl. M.	anderer Meinung/gleicher Meinung
BB	Der Betriebsberater
BG	Bundesgesetz
BGE	Entscheidungen des Schweizerischen Bundesgerichtes
BGr	Bundesgericht
BGH	Bundesgerichtshof
Bull. Cr. Soc.	Bulletin of the Copyright Society of the USA
Bull.	Bulletin
CBEMA	Computer and Business Equipment Manufacturers Association (USA)
Comp. Bull.	Computer Bulletin
CONTU	The National Commission on New Technological Uses of Copyrighted Works of the USA
DB	Der Betrieb
DdA	Le Droit d'Auteur
DIN	Deutsche Industrie-Norm
EDV	Elektronische Datenverarbeitung
Fn	Fussnote
GRUR	Gewerblicher Rechtsschutz und Urheberrecht
GRUR Int.	Gewerblicher Rechtsschutz und Urheberrecht, Internationaler Teil
IBM	International Business Machines Corp.
IIC	International Review of Industrial Property and Copyright Law
INTERGU	Internationale Gesellschaft für Urheberrecht e.V.
Mitt.	Mitteilungen
NCR	NCR Corp.
n.F.	neue Folge
NJW	Neue Juristische Wochenschrift
NZZ	Neue Zürcher Zeitung
OLG	Oberlandesgericht
OMPI	Organisation mondiale de la propriété intellect.
OR	Schweizerisches Obligationenrecht
Rev.	Review
SJZ	Schweizerische Juristenzeitung
Schw. Mitt.	Schweizerische Mitteilungen für gewerblichen Rechtsschutz
UFITA	Archiv für Urheber-, Film-, Funk- und Theaterrecht
VE	Vorentwurf
Vol.	Volume
WIPO	World Intellectual Property Organization
ZBJV	Zeitschrift des Bernischen Juristenvereins
ZGB	Schweizerisches Zivilgesetzbuch
ZSR	Zeitschrift für Schweizerisches Recht

Literaturverzeichnis

Die nachstehend aufgeführten Werke werden nur mit dem Namen des Autors, wo nötig noch mit einem Stichwort zitiert. Falls ein Autor mit mehreren Werken nur mit seinem Namen zitiert wird, ist das erste der unter seinem Namen aufgeführten Werke gemeint. Weitere Literaturangaben zu Einzelfragen und zu ausländischen Stellungnahmen finden sich in den einzelnen Anmerkungen.

ABA (The American Bar Assoc.) Symposium 1967 (Diverse Autoren): Computers and Copyright, Bull. Copyright Society of the U.S.A., Vol. 15, 1967/68, S. 1

AIPPI (International): Diskussion zur Frage 57 B betreffend den Schutz von Computerprogrammen u.a. in GRUR Int. 1974, S. 128; GRUR Int. 1975, S. 349; GRUR Int. 1978, S. 441; AIPPI-Jahrbücher 1971ff., insbesondere 1973/I, S. 401-442

AIPPI (Schweizergruppe): Zur Frage 57 B u.a. in Schw. Mitt. 1973, S. 16f (FINSLER CH.), Jahrbücher 1973ff, diverse Protokolle, Schr. Manuskript vom 3. 4. 75; Schw. Mitt. 1976, S. 133 (KIRKER G. E.); Schw. Mitt. 1978, S. 142 (KIRKER G. E.)

AXTER H./AXTER O.: Die Urheberrechtsschutzfähigkeit von Rechenprogrammen, Der Betriebsberater 1967, S. 606

BANZHAF J. F.: Computers and the Copyright Law: A Commentary, ACM Communications No. 1 (1967), S. 61

Ders.: Copyright Protection for Computer Programs – Ascap Copyright Law Symposium (1966), S. 188

BENDER D.: Trade Secret Software Protection, in Software Protection by Trade Secret, Contract, Patent Law, Practise and Forms, Washington D.C. 1969 (Patent Resources Group), S. 3

BIGELOW R. P.: The Patentability of Software, Modern Data, June 1972, S. 34

BLUM R.: Schutz der Immaterialgüter vor sklavischer Nachahmung auf technischem Gebiet, Diss. Zürich 1947

Ders.: Sind Computerprogramme patentfähig?, NZZ Nr. 568 vom 6. Dezember 1973, S. 35

BOHR K.: Die Urheberrechtsbeziehungen der an der Filmherstellung Beteiligten, UFITA Bd. 57, 1977, S. 1

DE BRABANTER M./VANDERPERRE R.: Aspects particuliers de la protection juridique du logiciel, L'ingénieur-conseil (Bruxelles), 1976, S. 137

BRAUN B.: Rechtsschutz von Computerprogrammen, Der Betriebsberater, 1971, S. 1343

BREYER ST.: The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies and Computer Programs, Harvard Law Review, Vol. 84, 1970, S. 281

The British Computer Society: Software Protection. The Legal Protection of Computer Programs (Proceedings of a Conference at Brighton), 1970

BRUTSCHKE P.-G.: Urheberrecht und EDV, München 1972

CARNAHAN W. H.: Protection of Computer Programs – A Dilemma, Research Report 72-7 der United States Air Force Academy, Colorado Springs 1972

CARY GEORGE D.: Copyright-Eintragung und Rechenprogramme, GRUR Int. 1964, S. 623

CBMA (Computer and Business Machine Assoc.): CBMA Position on Copyright Revision Bill to the House Judiciary Committee, Interne Veröffentl. der IBM

CONTU (National Commission on New Copyrighted Works): The Commission and its Recommendations, Performing Arts Review Vol. 9 (1979), S. 32

Copyright-Office USA: Zur Copyright-Eintragung von Rechenprogrammen in den USA, GRUR Int. 1964, S. 635

DANIELS A./YEATES D./ERBACH K.: Grundlagen der Systemanalyse, Zürich 1971

- DIERSTEIN/FIEDLER/SCHULZ (Hrsg.): Datenschutz und Datensicherung (Fachtagung 1976), Köln 1976
- DITTRICH R.: Elektronische Datenverarbeitung und Urheberrecht, in Österr. Bl. für gewerbli. Rechtsschutz und Urheberrecht, 1970, S. 1
- DWORATSCHEK S.: Einführung in die Datenverarbeitung, 5. Aufl. Berlin/New York 1973
- ELBEN W.: Technische Lehre und Anweisung an den menschlichen Geist im Patent- und Gebrauchsmusterrecht, Diss. Frankfurt a.M. 1960
- FABIANI F.: Sind Apparate geistige Schöpfer?, GRUR Int. 1965, S. 422
- FINSLER CH.: Der Schutz von Computerprogrammen, Schw. Mitt. 1973, S. 16
- FUCHS A.: Der Erwerb von Computern (Vertragstypen und besondere Probleme der Gewährleistung), Diss. Zürich 1978
- FURRER R.: Automatische Datenverarbeitung, Zürich 1972
- GALBI E.: Proposal for New Legislation to Protect Computer Programming, Bull. Copyright Society of the USA Vol. 17, 1969/70, S. 280
- GAMM O. FRH. VON: Der urheber- und wettbewerbsrechtliche Schutz von Rechenprogrammen, in Wettbewerb in Recht und Praxis 1969, S. 96
- GANZHORN K./TJADEN K.: Was ist Software?, Zürich 1970
- GERNERT D.: Einführung in die Datenverarbeitung für Juristen, München 1974
- GIRTH P.: Individualität und Zufall im Urheberrecht, Berlin 1974
- Ders.: Der künstlerische Aspekt des Computers – eine urheberrechtliche Betrachtung, in Film und Recht, 1974, S. 433
- GOLDBERG D.: Legal Protection of EDP Software, Datamation, May 1972, S. 66
- GOTZEN F.: Le droit d'auteur face à l'ordinateur, DdA 1977, S. 15
- Ders.: L'ordinateur et la propriété intellectuelle, Journal des Tribunaux (Bruxelles), Februar 1976, S. 86
- HARTMANN G.: Praxis der elektronischen Datenverarbeitung, Berlin 1970
- HASS P.: Gedanken zur sklavischen Nachahmung, GRUR 1979, S. 361
- HEIM E./CURCHOD F.: La protection des programmes d'ordinateur, Schw. Mitt. 1972, S. 103
- HEIM E.: Die statistische Einmaligkeit im Urheberrecht de lege lata und de lege ferenda, Diss. Freiburg 1971
- HELBER F.: Die wissenschaftlichen und technischen Leistungen und der urheberrechtliche Werkbegriff, Diss. Basel 1968
- HOEPFFNER G.: Warenzeichenrechtlicher Schutz von Namen für Datenverarbeitungsprogramme, GRUR 1974, S. 546
- HUBER A.: Computer-Programme und Software-Engineering in patentrechtlicher Sicht, Mitt. der deutschen Patentanwälte 1975, S. 101
- HUBMANN H.: Urheber- und Verlagsrecht, 4. Aufl. München 1978
- IBM (UK): Copyright in Computer Output, Internes Memorandum vom 22. 5. 75
- Institut für Automation und Operations Research der Universität Freiburg: Bestand an Computern Schweiz 1978, Freiburg/Schweiz 1978
- ISELIN F.: Computerprogramm und Technik, Diss. Basel 1975
- ISIS Infratest Information Services: Computerprogramme in Deutschland, Österreich, Schweiz, München 1978
- JANSEN D.: Der Markt für Computerdienstleistungen in Deutschland, Diss. Hamburg 1972
- KATZENBERGER P.: Urheberrecht und Dokumentation, GRUR 1973, S. 629

- KIENZLE O.: Die Patentierbarkeit von Computerprogrammen, Berlin 1975
- KINDERMANN M.: Special Protection Systems for Computer Programs – A Comparative Study, IIC Vol. 7, 1976, S. 301
- Ders.: Zur patentrechtlichen Grenzziehung zwischen Rechenregel und technischer Erfindung, GRUR 1974, S. 305
- Ders.: Die Rechenvorschrift im Patentanspruch, GRUR 1969, S. 509
- KIRBY P.: Industrial Property Protection for Software, IIC Vol. 5, 1974, S. 169
- KIRKER G. E.: Protection des programmes d'ordinateur, Schw. Mitt. 1976, S. 133
- KLINTE E.: Software-Vermarktung – eine Urheberrechtsfrage, in Online-adl-Nachrichten, Nr. 3, 1978, S. 174
- KNAP K.: Künstlerisches und wissenschaftliches Werk als Schutzobjekt des Urheberrechts, in Festschrift für TROLLER A., Basel und Stuttgart 1976, S. 117
- KÖHLER R.: Der urheberrechtliche Schutz der Rechenprogramme, Diss. München 1968
- Ders.: Zum Urheberschutz für Programme elektronischer Datenverarbeitungsanlagen, Der Betriebsberater 1969, S. 1114
- KOKAI T. R.: Rechtsschutz für Computersoftware, Data Exchange 7/8 (1975), S. 11
- KOLLE G.: Der Rechtsschutz von Computerprogrammen aus nationaler und internationaler Sicht, GRUR 1973, S. 611, und GRUR 1974, S. 7
- Ders.: Technik, Datenverarbeitung und Patentrecht (Bemerkungen zur Dispositionsprogramm-Entscheidung des BGH), GRUR 1977, S. 58
- Ders.: Besprechung von SIDLER P., Diss. Bern 1968, GRUR Int 1969, S. 457
- Ders.: Schutz der Computerprogramme, Bericht der Deutschen Landesgruppe AIPPI, GRUR Int. 1974, S. 129, und GRUR 1974, S. 448
- Ders.: Présent et avenir de la protection du logiciel, DdA 1977, S. 69
- KUMMER M.: Das urheberrechtlich schützbares Werk, Bern 1968
- Ders.: Die Entgrenzung der Kunst und das Urheberrecht, in Festschrift für TROLLER A., Basel und Stuttgart 1976, S. 89
- LAWLOR R. C.: Copyright Aspects of Computer Usage, Communications of the ACM (Assoc. for Computing Machinery), Vol. 7, No. 10, 1964, S. 572
- Ders.: The American View, Brighton Conference 1969, Report, S. 22
- LARESE W.: Werk und Urheber in der schweizerischen Urheberrechtsreform, UFITA Bd. 79, 1977, S. 153
- Ders.: Aktuelle Fragen zum urheberrechtlichen Werkbegriff, Film und Recht 1978, S. 74
- Ders.: Rezensionsabhandlung zu PURTSCHERT J., Diss. Freiburg 1974, UFITA Bd. 75, 1976, S. 377
- LEHMANN B.: EDV-Programme als Vertragsgegenstand, sysdata und bürotechnik, Folgen 5–12, Zürich 1978
- Ders.: Der Rechtsschutz von Rechenprogrammen und der Lizenzvertrag für IBM-Programmprodukte, Interne Studie der IBM (Schweiz), 8. 11. 1972
- LÖBEL/MÜLLER/SCHMID, EDV Lexikon, 5. Aufl. München 1970
- LUTZ H.: Lizenzierung von Computerprogrammen, GRUR 1976, S. 331
- LUTZ M. J.: Die Schranken des Urheberrechtes nach schweizerischem Recht unter vergleichender Berücksichtigung der europäischen Urheberrechtsgesetze und der deutschen Entwürfe, Diss. Zürich 1964
- MARMOREK E.: Schutz und Lizenzierung von Know how in den USA, Österr. Bl. 1976, S. 96

- MERTENS P.: Angewandte Informatik, Berlin/New York 1972
- MÖHRING PH.: Die Schutzfähigkeit von Programmen für Datenverarbeitungsanlagen, GRUR 1967, S. 269
- Ders.: Können technische, insbesondere Computererzeugnisse Werke der Literatur, Musik und Malerei sein?, UFITA Bd. 50, 1967, S. 835
- MÖHRING/NICOLINI: Urheberrechtsgesetz, Kommentar, Berlin und Frankfurt 1970
- MOOERS C. N.: Computer Software and Copyright, ACM Computing Surveys, Vol. 7, No. 1, 1975, S. 45
- MOSER H.: Vergleichende Literaturuntersuchung hinsichtlich der Schutzfähigkeit von Rechenprogrammen für Datenverarbeitungsanlagen, GRUR 1967, S. 639
- MÜLLER P.: Lexikon der Datenverarbeitung, 5. Aufl. München 1972
- MYERS E. D.: Should Software be Copyrighted?, Datamation, March 1978
- NEUGEBAUER U./DEHN D./PHILIPP U./WALTER H. E., Stand und Entwicklung des Marktes für mehrfach verwendbare Anwendungssoftware in der gewerblichen Wirtschaft, Forschungsbericht DV 76-04 des Bundesministeriums für Forschung und Wirtschaft, München 1976
- NEUMEYER F.: Zum Schutz von Computerprogrammen im schwedischen Recht, GRUR Int. 1966, S. 417
- NEUMEYER F.: Zum Schutz von Rechenprogrammen im englischen Recht, GRUR Int. 1966, S. 522
- NORDEMANN W./ROEBER G. (Hrsg.): Das neue U.S. Copyright Law, (erläutert von europäischen und amerikanischen Juristen), Berlin 1978
- NYCUM S. H.: Legal Protection of Proprietary Rights in Software, (National Science Foundation), Washington D.C. 1976
- OEHLSCHLEGEL H.: Sollen und können Rechenprogramme geschützt werden?, GRUR 1965, S. 465
- Ders.: Zur Schutzfähigkeit von Rechenprogrammen für Datenverarbeitungsanlagen, GRUR 1968, S. 679
- OELSCHLÄGER C.: Einführung EDV (Skriptum), München 1977
- PAGENBERG B. A.: Patentability of Computer Programs on the National and International Level, IIC, Vol. 5, No. 1, 1974, S. 1
- PARISINI/WÄCHTER: Organisationshandbuch für die Einführung von ADV-Systemen, Berlin 1971
- PEDRAZZINI M.: Über den Leistungsschutz der Interpreten, der Ton- und Tonträgerhersteller und der Sendeunternehmen, ZSR 96 n.F. (1977), II, S. 1
- Ders.: Hauptakzente der schweizerischen Revision des Urheberrechtsgesetzes, UFITA Bd. 50, 1967, S. 915
- Ders.: Zur Revision des schweizerischen Urheberrechts, in Film und Recht Nr. 2, 1973, S. 67
- Ders.: Skriptum Immaterialgüterrecht, St. Gallen 1979
- PERRET F.: La protection des prestations en droit privé suisse, ZSR 96 n.F. (1977), II, S. 199
- PLANDER H.: Wissenschaftliche Erkenntnisse und Urheberrecht an wissenschaftlichen Werken, UFITA Bd. 79, 1976, S. 25
- PORTIN G.: Zum Schutz von Computerprogrammen in Finnland, GRUR Int. 1975, S. 19
- PRASINOS N.: Worldwide Protection of Computer Programs by Copyright, Report Honeywell Corp. o. J.
- PURTSCHERT J.: Der Schutz des unmittelbaren Ergebnisses einer Arbeits- und Unternehmerleistung, Diss. Freiburg i. Ue. 1974
- RAPP K.: Der Anwendungsbereich der Rechte zur Verbreitung und zur körperlichen Wiedergabe im schweizerischen Urheberrecht, Basel 1974

RAU G.: Antikunst und Urheberrecht, Berlin 1978

RAU O.: Das Verhältnis des § 1 Wettbewerbsgesetzes zu den Spezialnormen des gewerblichen Rechtsschutzes, insbesondere unter dem Gesichtspunkt der Ausnutzung fremder Arbeitserzeugnisse, Diss. Erlangen 1931

REHBINDER M.: Recht am Arbeitserzeugnis und Urheberrecht, in Festschrift für ROEBER G., UFITA Bd. 66, 1973, S. 125

REICHEL W.: Die Formgebung und die Formgestaltung von Schriftwerken und ihr rechtlicher Schutz, GRUR 1963, S. 124

RINGER B.: Speicherung und Wiedergabe urheberrechtlich geschützter Werke durch Datenverarbeitungsanlagen, GRUR Int. 1968, S. 18

SEIPEL P.: Software Protection and Law, Data No. 6, 1975, S. 44

SIDLER P.: Der Schutz von Computerprogrammen im Urheber- und Wettbewerbsrecht, Diss. Bern 1968

SIEGWART H.: Der urheberrechtliche Schutz der wissenschaftlichen Werke, Diss. Bern 1953

SKAUPY W.: Möglichkeiten des Urheber-, Patent- und Wettbewerbsschutzes von elektronischen Rechenprogrammen, Mitt. Patentanwälte 1967, S. 121

SOLTYSINSKI ST.: Computer Programs and Patent Law: A Comparative Study, Rutgers Journal of Computers and the Law No. 1, 1973

SCHENK U.: Der urheberrechtliche Werkbegriff unter besonderer Berücksichtigung der Probleme der «Neuen Musik», Diss. Freiburg i. Ue. 1977

SCHMIEDER H. H.: Zur Schutzfähigkeit von Werken und Leistungen zweiter Hand, Film und Recht, 1978, S. 84

Ders.: Die sogenannte Verwandtschaft zwischen Leistungsschutz und Urheberrecht, Film und Recht Nr. 1, 1972, S. 22

SCHNEIDINGER C.: Der Leistungsschutz unter besonderer Berücksichtigung der technischen Leistung, Diss. St. Gallen 1977

SCHORN F.: Zum Leistungsschutz nach deutschem Recht, GRUR 1978, S. 230

SCHRÖDER O.: Rechtsschutz von Computerprogrammen, Referat vom 13. 4. 73 in Rüschklikon ZH (Protokoll), S. 1

SCHULZ A.: Informatik für Anwender, Berlin/New York 1973

SCHWAIGER H./KOCKLER F. J.: Zum Rechtsschutz und zur Verkehrsfähigkeit von Rechenprogrammen, Wirtschaftsrecht Nr. 3, 1973, S. 353

Schweizerischer Juristenverein: Protokoll der 111. Jahresversammlung des Schweizerischen Juristenvereins vom 23. – 25. September 1977, ZSR 96 (1977), S. 393

STROH R.: Werkeinheit und Werkmehrheit im Urheberrecht, Diss. München 1969

THURMANN R.: Der Rechtsschutz von Computerprogrammen in den USA, GRUR Int. 1969, S. 207

TROLLER A.: Immaterialgüterrecht, Bd. 1 und Bd. 2, 2. Aufl. Basel und Stuttgart 1968/1971

Ders.: Kurzlehrbuch des Immaterialgüterrechts, Basel und Stuttgart 1973

Ders.: Urheberrecht und Ontologie, UFITA Bd. 50, 1967, S. 385

Ders.: Der urheberrechtliche Schutz der technischen Zeichnung, SJZ 60, 1964, S. 369

Ders.: Rapport zu Handen der Mitglieder der Schweizergruppe der AIPPI betreffend die Wiedergabe von technischen Zeichnungen, architektonischen Plänen, das Nachgiessen von Industrie-Erzeugnissen und ähnliche unmittelbare Benutzungen der Arbeitsleistungen, die im Rahmen eines Unternehmens erbracht werden, Schw. Mitt. 1964/65, S. 212

Ders.: Vorentwurf der Expertenkommission für ein schweizerisches Bundesgesetz betreffend das Urheberrecht, die leitenden Ideen, Schriftenreihe der InterGU Band 48, München 1972

- Ders.: Besprechung des Werkes von KUMMER M., Das urheberrechtlich schützbares Werk, ZBJV 1968, S. 232
- Ders.: Ist der immaterialgüterrechtliche numerus clausus der Rechtsobjekte gerecht?, in Ius et Lex, Festgabe für GUTZWILLER M., Basel 1959, S. 769
- TRÜEB H.: Der urheberrechtliche Schutz wissenschaftlicher Werke, Schw. Mitt. 1958/59, S. 58
- ULMER E.: Urheber- und Verlagsrecht, 2. Aufl. Berlin/Göttingen/Heidelberg 1960
- Ders.: Der Urheberschutz wissenschaftlicher Werke unter besonderer Berücksichtigung der Programme elektronischer Rechenanlagen, München 1967
- Ders.: Der urheberrechtliche Werkbegriff und die moderne Kunst, Rezensionsabhandlung zu KUMMER M., Das urheberrechtlich schützbares Werk, GRUR 1968, S. 527
- Ders.: Elektronische Datenbanken und Urheberrecht (Gutachten im Auftrag der Unesco und WIPO), München 1971
- Ders.: Einspeicherung und Wiedergewinnung urheberrechtlich geschützter Werke durch Computer-Anlagen, GRUR 1971, S. 297
- Ders. und KOLLE G.: Einspeicherung geschützter Werke in automatische Informations- und Dokumentationssysteme, GRUR 1976, S. 108
- WALTER M.: Zur Revision des österreichischen Urheberrechts, GRUR Int. 1974, S. 429
- WASSNER F.: Kunst, Geschmack und unlauterer Wettbewerb (Der Schutz der Form im Urheber-, Geschmacksmuster- und Wettbewerbsrecht), UFITA Bd. 52 1975, S. 1
- WEICHSELBAUMER C.: Methoden und Hilfsmittel zur Entwicklung von Benutzersprachen, in Data Report (Siemens), Heft 3/1979, S. 40
- WEISSTHANNER M.: Urheberrechtliche Probleme neuer Musik, Urheberrechtl. Abhandl. des Max Planck-Institutes für ausl. und intern. Patent-, Urheber- und Wettbewerbsrecht, Heft 14, München 1974
- WIPO: Model Provisions on the Protection of Computer Software, WIPO-Publ. No. 814 (englische und französische Ausgabe), Genf 1978, (teilweise in deutscher Übersetzung in GRUR Int. 1978, S. 286)
- WIPO: Report: Meeting Working Group on Copyright Problems arising from the Use of Computers, Genf 1979
- WITTE A.: Mikroprozessoren und Patentschutz, GRUR 1978, S. 511
- ZAHN R.: Urheberrecht und Computer-Programme – Versuch einer Bestandsaufnahme, GRUR 1978, S. 207
- ZELLE K./SCHLECHTNER O./SCHMID F.: Algorithmen, Programme, Programmpakete – Eine Programm-bibliothek für Wissenschaft und Verwaltung, Wien 1975
- ZIMMERMANN W. L.: Datenverarbeitung – Lehrbuch der Betriebsinformatik, Bd. 1 und 2, Wiesbaden 1975 und 1977
- ZIPSE E.: Sind Computerprogramme Anweisungen an den menschlichen Geist?, GRUR 1973, S. 123
- Ders.: Wird das künftige europäische Patenterteilungsverfahren den modernen, zukunftsintensiven Technologien gerecht?, GRUR Int. 1973, S. 185

Quellenverzeichnis

Bundesgesetz betreffend das Urheberrecht an Werken der Literatur und Kunst (vom 7. Dezember 1922) (zit.: URG)

Bundesgesetz über den unlauteren Wettbewerb (vom 30. September 1943) (zit.: UWG)

Bundesgesetz betreffend die Erfindungspatente (vom 25. Juni 1954) (zit.: PatG)

(deutsches) Gesetz über Urheberrecht und verwandte Schutzrechte (vom 9. September 1965) (zit.: UrhG)

Vorentwurf I des Bundesgesetzes betreffend das Urheberrecht und Vorentwurf II betreffend das Urheberrecht (zit.: VE I und VE II)

Entscheidungen des schweizerischen Bundesgerichtes und des deutschen Bundesgerichtshofes

Revidierte Berner Übereinkunft zum Schutze von Werken der Literatur und Kunst vom 9. September 1886 (zit.: RBUE)

Welturheberrechtsabkommen vom 6. September 1952 (zit.: WUA)

Model Provisions on the Protection of Computer Software der WIPO, WIPO-Publ. 814, englische Ausgabe, Genf 1978 (zit.: MV)

1. Kapitel: Einführung in die Problematik und Vorgehen

A. Relevanz und Vorgehen

Als Vorfrage muss die Bedeutung des urheberrechtlichen Schutzes oder eines de lege ferenda zu schaffenden Sonderschutzes in dreifacher Hinsicht untersucht werden:

1. Mit Blick auf die Entwicklungen in der Softwareherstellung und -vermarktung unter Berücksichtigung der davon berührten Interessengruppen¹.

2. Mit Blick auf die noch sehr entwicklungsfähigen technischen Schutzmöglichkeiten und weitere ausserrechtliche Hemmnisse für eine unberechtigte Verbreitung von Software².

3. Mit Blick auf die Tauglichkeit bestehender rechtlicher Schutzmöglichkeiten, vor allem des Vertragsrechts³, des Wettbewerbsrechts⁴ und des Patentrechts⁵.

Hauptaufgabe dieser Arbeit ist dann die Beurteilung des urheberrechtlichen Schutzes unter Berücksichtigung der gegenwärtigen Reformvorstellungen⁶. Angesichts der Zweifel und Ungereimtheiten im Urheberrecht werden anschliessend die Bestrebungen um einen Sonderschutz de lege ferenda erörtert⁷.

Aus Gründen der Übersichtlichkeit und wegen ihrer Besonderheiten und verhältnismässig geringen praktischen Bedeutung werden Softwareentwicklungen, die auf die Erzeugung von künstlerischen bzw. anderen als kommerziellen und technisch-wissenschaftlichen Ergebnissen ausgerichtet sind, in dieser Arbeit nicht berücksichtigt⁸.

B. Einführung in den Stand der Diskussion

Trotz bereits fünfzehnjähriger reger Erörterung gilt noch heute die von KOLLE im Jahre 1973 getroffene Feststellung, dass «viele der mit dem Schlagwort ‹Schutz von Computerprogrammen› aufgeworfenen Fragen

1 Unten Kap. 3 A, 3 C.

2 Unten Kap. 3 B.

3 Unten Kap. 4 B.

4 Unten Kap. 4 C.

5 Unten Kap. 4 D.

6 Unten Kap. 5 und 6.

7 Unten Kap. 7.

8 Informativ zu kunsterzeugenden Computerprogrammen und deren Erzeugnisse u.a. GIRTH P., S. 80ff; ders., in Film und Recht 1974, S. 433ff; WEISSTHANNER M., S. 87ff; KUMMER M., Werk, S. 176ff.

schwierig, die meisten umstritten und noch ungeklärt sind»⁹. Immerhin konnte inzwischen – mit Ausnahme des Urheberrechtes – über einiges weitgehende Einigung erzielt werden¹⁰: Es bleibt den Software-Herstellern nur übrig, «für ihre Softwareprodukte, namentlich maschinenlesbare Programme nebst zugehöriger Dokumentation, auf den Urheberschutz und den in Zukunft vielleicht verfügbaren Sonderschutz für Software zu vertrauen»¹¹. Ideale Lösungen gibt es weder mit Hilfe des gewerblichen Rechtsschutzes noch im Urheberrecht¹², und auch der Spielraum für einen urheberrechtlichen Schutz ist sehr klein und wird unterschiedlich bewertet¹³.

9 KOLLE G., GRUR 1973, S. 612.

10 Vgl. unten Kap. 4. Es sei vorweggenommen, dass die vertragsrechtliche Situation klar ist, im Patentrecht aufgrund höchstrichterlicher Rechtsprechung in allen kontinental-europäischen Ländern kaum mehr Raum für einen Softwareschutz besteht, die wettbewerbsrechtliche Diskussion noch kontrovers, im Ergebnis aber nicht vielversprechend ist.

11 KOLLE G., GRUR 1977, S. 74: So das Fazit im Anschluss an die Besprechung der Dispositionsprogramm-Entscheidung des BGH (vgl. unten Fn 12 b). Vgl. auch den Hinweis des BGH auf die Möglichkeit eines Urheberschutzes. Der Entscheid wurde aber offengelassen.

12 Vgl. unter vielen anderen neueren oder wichtigeren Publikationen:

a) *Allgemein*: KOLLE G., GRUR 1973, S. 612ff und 1974, S. 7ff; KOLLE G., DdA 1977, S. 69ff; KINDERMANN M., IIC 1976, Vol. 7, S. 301ff; HEIM/CURCHOD, Schw. Mitt. 1972, S. 103ff; KIRKER G. E., Schw. Mitt. 1976, S. 133ff; PERRET F., ZSR 96 (1977), S. 203ff.

b) *Patentrecht*:

Inland: Verfüg. des Amtes für geistiges Eigentum vom 3. 9. 68, Aktz.: 3055/67 = SchPMMBI. 1968 I 53 = GRUR Int. 1969, S. 142 (Bewehrungs-Rechenprogramm); BGE 95 I 579 (Energieverteilungsprogramm); BGE 98 Ib 396 = SchwPMMBI. 1973, S. 26 (Dimensionale Synthese); ferner auch ISELIN F., Diss. Basel 1975.

Ausland:

BRD: KOLLE G., GRUR 1977, S. 58ff, Bemerkungen zur wegweisenden Dispositionsprogramm-Entscheidung des BGH, Aktz.: X ZB 23/74 = GRUR 1977, S. 96ff, ferner auch Urteil des Bundespat. Ger., 17. Senat, vom 19. 1. 78, Aktz.: 17 W (pat) 53/73 = GRUR 1978, S. 705; ferner BGH in GRUR 1977, S. 657; BGH in GRUR 1978, S. 102; KIENZLE O., Die Patentierbarkeit von Computerprogrammen, Berlin 1975.

Zur Problematik i.w.S.: KINDERMANN M., GRUR 1969, S. 509 und GRUR 1974, S. 305 betreffend Rechenregel und Patentanspruch. Internationale Übersichten: PAGENBERG B. A., IIC 1974, Vol. 5, No. 1, S. 1ff; KOKAI T. R., Data Exchange 1975, S. 14ff; LEHMANN B., Sysdata 5/1978, S. 2ff; KIENZLE O., S. 78ff; SOLTYSINSKI ST., Rutgers Journal of Computers and the Law, 1973, Vol. 3, No. 1, S. 1ff.

Urteile aus diversen Ländern: Österreich: Entsch. des Österr. Patentamtes vom 12. 12. 67 = GRUR Int. 1968, S. 211 (Zinszahlen-Rechenprogramm); Frankreich: Entsch. Cour de Paris vom 22. 5. 73 = GRUR Int. 1974, S. 279 (Pigmentauswahl); USA: u.v.a. GRUR Int. 1972, S. 128 (BCD-Umwandlung), GRUR Int. 1973, S. 75 (BCD-Umwandlung II), GRUR Int. 1978, S. 412 (Computersprachübersetzung), GRUR 1978, S. 465 (Alarmgrenzenanpassungs-Verfahren).

c) *Wettbewerbsrecht*:

SIDLER P., S. 49ff; SCHNEIDINGER C., S. 39ff; PURTSCHERT J., S. 109ff.

d) *Urheberrecht*:

TROLLER A., Ontologie, UFITA, Bd. 50, 1967, S. 414ff; ULMER E., Wiss. Werke, S. 3ff; SIDLER P., S. 1ff; KÖHLER R., S. 1ff; HEIM/CURCHOD und PERRET siehe Zitat unter lit. a; SCHWAIGER H./KOCKLER F. J., S. 353ff, und ZAHN R., GRUR 1978, S. 207ff, letztere als Bestandsaufnahmen zum umfangreichen Schrifttum.

13 Die schweizerische Lehre lehnt den urheberrechtlichen Schutz von Computerprogrammen fast einhellig ab. Im Gegensatz dazu steht die deutsche Lehre, die «nach

Woran liegt es wohl, dass die traditionellen Gebiete des Immaterialgüterrechtes für die neuartigen geistigen Erzeugnisse keine «adäquaten Lösungen»¹⁴ anbieten und der Ruf nach Schutzsystemen *sui generis*¹⁵ stärker wird?

Ungeachtet der Ziele¹⁶, die mit einem Schutz der Computersoftware verfolgt werden, und unabhängig von einer Bewertung der Interessen eines noch immer stark expandierenden, bedeutenden Wirtschaftszweiges ist die gegenwärtige Rechtsunsicherheit¹⁷ unbefriedigend. Es bedarf einer Klärung der Rechtslage¹⁸, denn ein Blick in die Standardverträge der wichtigsten Softwarehersteller zeigt, dass man neben dem relativen Schutz durch Vertrag auf eine weitergehende monopolartige Rechtsstellung vertraut¹⁹, auch wenn man sich in der Regel auf kein bestimmtes Schutzsystem bezieht. Es fällt aber auf, dass sich die Schutzerwartungen auf das Urheberrecht, das als Geschäftsgeheimnis geschützte Know-how sowie auf den allgemeinen wettbewerbsrechtlichen Schutz konzentrieren²⁰. Soweit bekannt, waren die schweizerischen Gerichte bisher nicht mit Fällen aus diesem Gebiet befasst.

überwiegender Meinung bejaht» (ULMER E., GRUR 1971, S. 297) oder wo nach zurückhaltenderer Beurteilung «eine Vielfalt der Ansichten» (ZAHN R., Bestandsaufnahme in GRUR 1978, S. 218) festzustellen ist.

14 KOLLE G., DdA 1977, S. 69.

15 Siehe die vergleichende Studie über die bereits bestehenden Vorschläge in KINDERMANN M., IIC 1976, Vol. 7, No. 3, S. 301ff, (unten Kap. 7); ferner die seit Jahren im Rahmen der AIPPI geführte Diskussion (Frage 57B), die in der Entschliessung ausmündete, ein Sonderschutzsystem auf der Basis der Vorschläge der WIPO (unten Kap. 7) in Erwägung zu ziehen: Siehe u.a. KIRKER G. E., Schw. Mitt. 1978, S. 142; AIPPI Annuaire 1978/II, S. 236; GRUR Int. 1978, S. 441; Stellungnahmen der Schweizergruppe u.a.: FINSLER CH., Schw. Mitt. 1973, S. 16ff; KIRKER G. E., Schw. Mitt. 1976, S. 133ff; Manuskript vom 3. 4. 75 im Hinblick auf den Kongress von San Francisco, 1975; ferner u.a. GRUR Int. 1974, S. 358; GRUR Int. 1975, S. 349; AIPPI Annuaire 1973/1, S. 399ff und andere Jahrbücher nach 1970, insbes. 1975.

16 Als Gründe für einen wirksamen Softwareschutz werden üblicherweise genannt: Absicherung von hohen Investitionen an Arbeit und Kapital, Schutz von ideellen und vermögenswerten Interessen der an der Softwareentwicklung Beteiligten, Vermeidung von aufwendigen Parallelentwicklungen und Verbesserung der Information über neue Softwareentwicklungen, weil der ausreichende Schutz die Gefahren einer Offenbarung vermindert und somit Anreize für die Verbreitung vermittelt.

17 Die WIPO umschreibt die Situation unter internationalem Blickwinkel in der Einleitung zu ihren Mustervorschriften (WIPO-Publ. 814E, S. 5 und Zitate Kap. 7) wie folgt: «However, there is at present a state of uncertainty as to the protection of computer software under various legal systems.»

18 Diese Arbeit will deshalb in erster Linie ein weiterer Beitrag zur Feststellung der schweizerischen Rechtslage de lege lata sein. Über die Gegenüberstellung der dabei gewonnenen Ergebnisse mit bereits vorgeschlagenen Sonderschutzsystemen hinaus wird nicht beabsichtigt, neue Möglichkeiten eines Software-Sonderschutzes zu erörtern.

19 Vgl. LEHMANN B., Sysdata 1978, insbes. Teil 11/12 betreffend Programm-Lizenzierung; LUTZ H., GRUR 1976, S. 331ff; Interne Aktennotiz der NCR (Schweiz) vom 1. 2. 80 betreffend Intellectual Property Implementation; Interne Aktennotiz der IBM (Schweiz) vom 8. 11. 72; IBM-Form 120–2083 und Z 120–2065; ferner vgl. Fn 20 unten.

20 LEHMANN B., Sysdata 1978, Teil 12/78, S. 1; LUTZ H., GRUR 1976, S. 332; KLINTER W., S. 174; FUCHS A., S. 130; Gründe für das späte und zurückhaltende Interesse an urheberrechtlichen Schutzmöglichkeiten bei MOOERS, S. 63ff.

C. Vorläufige Lokalisierung der Problematik

Der Versuch, den vielschichtigen Charakter der Computersoftware²¹ und die damit verbundene Schutzproblematik aufzuzeigen, soll in freier Anlehnung an KNAPS²² kurzgefasste «wesensmässige Einteilung der Ergebnisse der schöpferischen Leistung»²³ erfolgen.

Dieser Autor unterscheidet drei Gruppen:

1. «Ergebnisse, welche in der Entdeckung einer Realität bestehen, die in der Natur oder Gesellschaft zwar vorhanden, jedoch bisher noch nicht bekannt war.»²⁴ Es geht um das im vorliegenden Zusammenhang nicht mehr weiter interessierende Gebiet der Entdeckungen.

2. «Ergebnisse, die in der Lösung eines Problems bestehen, die zwar durch die objektiven Realitäten (oder bereits bestehenden Lösungen) vorbestimmt, als Lösung jedoch neu ist²⁵.» Hier geht es um den finalen Schöpfungsprozess. Ziel ist die nutzbringende Problemlösung, an der, weil nur sie interessiert und schützenswert ist, allenfalls ein Schutzrecht angeknüpft wird. Durch ein Schutzrecht honoriert werden diese Ergebnisse u.a., weil der Lösungsweg, d.h. die Brücke zwischen Vorgegebenem²⁶ und Erreichtem, schöpferisch²⁷ ist, nicht aber das durch die objektiven Realitäten bestimmte Ergebnis selbst. Die Rechtsordnung hat sich hier bei der Zuordnung mit Parallelschöpfungen und Prioritätsproblemen zu befassen, weil nicht auszuschliessen ist, dass zwei Menschen im Rahmen eines determinierten Schaffensvorganges zu gleichen Ergebnissen gelangen.

3. «Ergebnisse, die in der Schaffung eines neuen Gutes bestehen, das ein individueller Ausdruck des Schaffenden ist²⁸.» Diese sind nur dann möglich, wenn im Gegensatz zu Ziff. 2 nicht zwei Menschen eine gleiche Schaffensgrundlage haben können und wenn ausserhalb eines gemeinsamen, durch die objektiven Realitäten vorgegebenen Ergebnisses oder Zweckes für eine freie Entfaltung noch Spielraum bleibt. Zweck, hier gemeint als Grundlage des finalen Handelns im weitesten Sinn, kann darüber hinaus

21 Vgl. ausführliche Begriffsbestimmung unter Kap. 2. Vorläufig dazu OELSCHLÄGER C., S. 156: «Software sind alle programmtechnischen Komponenten, die zum Betrieb einer EDV-Anlage und zur Lösung eines Verarbeitungsproblems erforderlich sind.»

22 KNAP K., S. 122ff: Diese Einteilung wurde nur soweit ausgeführt, als es dem Autor für die Zwecke jener Arbeit dienlich war, das Wesen des urheberrechtlich geschützten Werkes im Rahmen der ganzen Immaterialgüterordnung unabhängig von einzelnen Lehrmeinungen herauszuarbeiten.

23 KNAP K., S. 122.

24 Ebd.

25 Ebd.

26 Vorgegeben können u.a. die Entdeckungen oder die ihr innewohnenden Anwendungsmöglichkeiten und Zwecke (vgl. Ziff. 1) sein.

27 Schöpferisch-individuell sind z.B. die Gedankengänge, die begangenen Irrtümer bei der Lösungsfindung, die Form von Hilfskonstruktionen, überraschenden Erwägungen, Phantasie, Intuition usw.

28 KNAPP K., S. 123.

auch alles sein, was auf die Verwirklichung einer Anweisung²⁹ zielt, also alles, was als Vorgabe für eine schöpferische Leistung in einem abstrakten Prinzip derart fassbar ist, dass es, wenn verschiedenen Menschen mitgeteilt, imperativ in deren Schaffen einfließen kann³⁰. Das Schöpfungsziel darf nie für zwei Menschen vollends in gleicher Weise geistig erfassbar werden, sonst würde die dem Urheberrecht fremde Möglichkeit entstehen, dass unabhängig voneinander zwei gleiche Ergebnisse geschaffen werden.

Die zweite Gruppe fällt grundsätzlich in die Zuständigkeit des gewerblichen Rechtsschutzes und die dritte Gruppe in jene des Urheberrechtes³¹. Innerhalb der zweiten Gruppe ist zu unterscheiden zwischen den vom Erfindungsschutz erfassten Ergebnissen mit technischem Nutzeffekt und den sonderrechtlich nicht erfassten Problemlösungen nicht-technischer Natur, die durch rein verstandesmässige oder körperliche Betätigung des Menschen verwertet werden können, ohne dass dabei unmittelbar Naturkräfte eingesetzt werden. Es geht um die freien Anweisungen an den menschlichen Geist³².

Die soeben dargelegten Unterscheidungen lassen für die Computersoftware die Brennpunkte der Diskussion wie folgt erkennen:

Der Computer wird für einen bestimmten Zweck eingesetzt³³. Dieser wird erreicht, indem die Software den selbsttätigen Ablauf des (maschinellen) Verfahrens regelt, das zur Lösung eines Problems führt³⁴. Die Software als Problemlösungsregel zur Steuerung eines maschinellen Problemlösungsprozesses muss also ihren wesensmässigen Schwerpunkt auf dem Gebiet des gewerblichen (bzw. bei der oben erwähnten zweiten Gruppe) haben. Weiter spricht der Umstand, dass der automatisierte Arbeitsprozess durch die von der Software organisierten Schaltvorgänge abgewickelt wird, für die Nähe einer technischen und damit in das Erfindungsrecht fallenden Erscheinung. Allenfalls kann auch eine freie Anweisung an den menschlichen Geist in Betracht gezogen werden, wenn man argumentiert, die Software sei eine

29 Gleichbedeutend mit Anweisung: Manier, Rezept, Verfahren, Stil (vgl. KUMMER M., Werk, S. 52) oder auch Regel, Methode, Formel, Sachgebundenheit (vgl. ULMER E., Wiss. Werke, S. 16 und 18).

30 Unschwer können wir hier das von KUMMER aus anderer, phänomenologischer Sicht gewonnene Erfordernis der Zweckfreiheit erkennen (vgl. KUMMER M., S. 70ff; KUMMER M., Entgrenzung, S. 94ff).

31 Vereinfachend wollen wir von der ambivalenten Natur des Muster- und Modellschutzes absehen.

32 TROLLER A., IG I, S. 65 und 171; TROLLER A., Numerus clausus, S. 779f.

33 Über die verschiedenen Verwendungszwecke vgl. ZIMMERMANN W. L., Bd. I, S. 66.

34 Beispiel: Der Zweck sei, mit Hilfe eines Computers auf rationelle Weise aufgrund von einzelnen Ausgangswerten bestimmte Buchhaltungsergebnisse zu erarbeiten. Durch eine Vielzahl von Schaltvorgängen, die u.a. nach den Regeln des verwendeten technischen Systems und des anzuwendenden Buchhaltungssystems vor sich gehen (Verfahrensregel, Problemlösung, Software) wird das Buchhaltungsergebnis (Lösung, Output), aufgrund der im Betrieb vorhandenen Ausgangswerte ermittelt (Problem, Input).

Handlungslehre, die nur technische Grössen und Bedingungen des Computersystems berücksichtigt, ohne verändernd auf diese einzuwirken, weil sie nur von einem vorbestehenden technischen System bestimmungsgemässen Gebrauch macht.

Weiter abseits erscheint auf den ersten Blick das Urheberrecht: Die Software, die konkrete Darstellung der Steuerungsanweisung für einen maschinellen Problemlösungsprozess, lässt aufgrund ihrer strengen Zweckbindung nicht die Spielräume vermuten, die für eine freie Entfaltung des Schöpfers erforderlich sind. Dies primär deshalb, weil ihr Adressat, die Maschine, selber im Umfeld von technischen und organisatorischen Realitäten steht.

Der schöpferische Mensch steht bei der Entwicklung und Gestaltung der Software gleichzeitig im Banne der Maschine und der für ihren zweckmässigen Einsatz zu verwirklichenden Regeln einer Problemlösung.

Eine Zuweisung in ein einziges Schutzsystem scheint kaum möglich zu sein. Lange wurden denn auch Lösungen diskutiert, welche die Software je nach dem angesprochenen Element verschiedenen Schutzsystemen zuweisen. Aus der inzwischen vorliegenden höchstrichterlichen Rechtsprechung³⁵ zu den immaterialgüterrechtlichen Aspekten der Software ergibt sich aber, dass im wesentlichen nur das Urheberrecht in Frage kommt.

35 Unten Kap. 4 D.

2. Kapitel: Grundlagen für die Charakterisierung der infragestehenden Schutzobjekte¹

A. Allgemeine Begriffe der Elektronischen Datenverarbeitung

Jeder Datenverarbeitung liegt zugrunde, dass Eingangsinformationen (Daten, Eingangsdaten, Input, Informationen) nach bestimmten Verarbeitungsvorschriften (Regeln, Anweisungen) mit Hilfe technischer Einrichtungen in Ausgangsinformationen (Output, Ausgabedaten) umgestaltet werden². Von elektronischer Datenverarbeitung (kurz: EDV, ADV, DVA) spricht man, wenn die technischen Einrichtungen (Hardware, Computer, Datenverarbeitungsmaschine)³ vorwiegend elektronische Bauelemente enthalten. Die Verarbeitungsvorschriften (Programm-, Anweisungsvorschriften, Software⁴) werden nicht als Daten bezeichnet, obwohl sie sich in ihrer Speicherform im Hauptspeicher⁵ (Zentralspeicher, Primärspeicher, Arbeitsspeicher, Memory) und den peripheren Geräten⁶ (Randeinheiten, peripheral units) nicht von den Daten unterscheiden.

Der Unterschied liegt darin, dass die Verarbeitungsvorschriften (Software) die Vorgänge in der Zentraleinheit⁷ (Central Processing Unit) des Computers steuern, während Daten als Input Gegenstand oder als Output Ergebnis dieser Umformung sind. Der Computer ist nur insoweit befähigt, aus vorgegebenen Daten irgendwelche Ergebnisse zu erzeugen, als er eine vom Menschen logisch strukturierbare Aufgabenstellung nach einer lückenlosen Folge von Anweisungen verarbeiten kann⁸. Das bringt zugleich die

1 Nachstehend nur die elementarsten EDV-technischen Zusammenhänge. Weiteres im Laufe der Arbeit. Vgl. die umfangreichen Darstellungen im Rahmen von juristischen Arbeiten: bei FUCHS A., S. 8–36; SIDLER P., S. 1–9; KÖHLER R., Diss., S. 5–24; BRUTSCHKE P. G., S. 11–31; ISELIN F., S. 1–36; LEHMANN B., Teil 5/78, S. 1ff; DITTRICH R., S. 1ff.

2 Vgl. u.a. HARTMANN G., S. 6ff; ZIMMERMANN W. L., Bd. I, S. 25ff; OELSCHLÄGER C., S. 9ff; MÜLLER P. (Hrsg.), Lexikon, S. 145ff. Die Literatur weist eine grosse Begriffsvielfalt auf. Synonyme werden in Klammern beigelegt.

3 Die physikalischen, maschinellen Teile einer Datenverarbeitungsanlage. Die Gesamtheit aller realtechnischen Einrichtungen.

4 Software wird im allgemeinen an ihrem Gegensatz, der Hardware, definiert: «Alles was nicht Hardware, jedoch für den Einsatz eines Computers erforderlich oder nützlich ist, wird zur Software gerechnet» (LUTZ H., GRUR 1976, S. 331).

5 Der Hauptspeicher nimmt die zur Verarbeitung kommenden Teile der Software und die Daten auf.

6 Sammelbegriff für alle systemabhängigen Geräte, die unter Kontrolle einer Zentraleinheit arbeiten und damit integrierter Bestandteil eines EDV-Systems sind. Es sind dies u.a. Magnetplattenspeicher, Magnetbänder, Drucker, Bildschirme usw.

7 Das Kernstück der EDV-Anlage. Sie steuert und kontrolliert die Ausführung der durch die Software vorgegebenen Operationen.

8 Wahl, Grundlagen eines Management-Information-Systems, IBM-Nachrichten 1969, S. 917.

«Begrenztheit» des Computers zum Ausdruck: Eine Aufgabenstellung wird erst «computergerecht», wenn es dem Menschen gelingt, die Aufgabe logisch durchzugestalten und den Lösungsweg eines Problems in eine eindeutige und endliche Zahl von Arbeitsregeln (Umformungsregeln) aufzugliedern, durch deren schematische Befolgung man alle Aufgaben eines bestimmten Typs lösen, d.h. gegebene Grössen in andere zielgerecht umarbeiten kann⁹.

Sobald den Umformungsregeln schliesslich durch die Programmierung¹⁰ eine für den Computer verständliche Form von Anweisungen gegeben wird, kann man von einem für die konkrete Datenverarbeitungsanlage geeigneten Algorithmus sprechen oder von einem Programm¹¹, d.h. von einer auf dem Algorithmus beruhenden Anweisung für den Betrieb der EDV-Anlage. Das Programm ist die durch den jeweiligen Algorithmus vorgegebene notwendige Umsetzung und Zergliederung in Steueranweisungen, die letztlich im Befehlsrepertoire eines Prozessors (Hardware) eine Entsprechung haben.¹²

B. Was ist Software?¹³

I. Grundlagen

1. Das Verhältnis von Software und Hardware

Software und Hardware lassen sich nur bei einem bestimmten Computertyp genau abgrenzen. Funktionen oder Funktionsabläufe können

9 Eine «eindeutige Lösungsvorschrift» dieser Art wird als Algorithmus bezeichnet (vgl. MÜLLER P., S. 39; HARTMANN G., S. 6; ausführlicher ZIMMERMANN W. L., Bd. II, S. 173ff; unten Kap. 2 D im einzelnen). Der Algorithmus umfasst das allgemein definierte, programmierbare Verfahren zur Lösung einer bestimmten Aufgabe. Meistens sind im Algorithmus Verfahrensregeln zweier Ebenen zu einer untrennbaren Einheit verbunden: Einerseits solche des für den EDV-Einsatz bestimmten Aufgabengebietes (z.B. Buchhaltungsmethoden, statistische Verfahren) und andererseits EDV-technische Komponenten, die im Hinblick auf die Verwendung eines konkreten Datenverarbeitungssystems hinzugefügt wurden.

10 Programmierung ist der Arbeitsvorgang zur Herstellung eines Programmes (vgl. unten Fn 11). Das Programm wird in einer späteren Phase der Software-Entwicklung hergestellt (vgl. unten Kap. 2 C). Dieser Vorgang wird im engsten Sinne oft als Kodieren bezeichnet.

11 Das Programm ist das wesentlichste Element der Computersoftware (vgl. Definitionen unten Kap. 2 B I 2). Nachstehend wird vorläufig stets vom codierten Programm gesprochen als der bereits in eine vom Computer interpretierbare Form gebrachten Befehlsabfolge.

12 Vgl. unten Kap. 2 B I 2.

13 Näheres bei GANZHORN K./TJADEN K., S. 6ff; DWORATSCHEK S., S. 287ff; MÜLLER P. (Hrsg.), S. 519f; ZIMMERMANN W. L., Bd. II, S. 255ff.

Die Literatur macht im Zusammenhang mit Software noch weitere, für unsere Betrachtungen entbehrliche Unterscheidungen: Brainware (intellektuelle Dienstleistungen), Orgware (EDV-gerechte Organisation eines Betriebes); Firmware (Mikroprogramme).

nämlich sowohl von der Software als auch von der Hardware gesteuert werden. Sie sind also bis zu einem gewissen Grad austauschbar. Aufgrund der Fortschritte in der Entwicklung von Prozessoren und Festspeichern neigt man heute auch bei Universalcomputern immer mehr dazu, häufig wiederkehrende Grundfunktionen hardwaremässig auszustatten¹⁴. Anders als bei den determinierten, d.h. den auf einen beschränkten Aufgabenkreis festgelegten Computersystemen, wird aber im wichtigeren Bereich der Universalcomputer aus Gründen der Flexibilität die Software – in künftig vielleicht etwas reduziertem Umfange – unentbehrlich bleiben, um die vorhandenen Hardwarefunktionen im Rahmen der Leistungsmöglichkeiten für die Lösung beliebiger Datenverarbeitungsaufgaben zu koordinieren.

2. *Elemente und Grundtypen von Programmen*

Eine EDV-gerechte Lösungsvorschrift muss in einem Programm¹⁵, in einer eindeutigen und lückenlosen Folge von Befehlen¹⁶, angeordnet werden.

Erfahrungsgemäss werden beim Programmieren auf einem bestimmten Arbeitsgebiet in Wirtschaft, Technik und Wissenschaft wiederkehrend gleiche, teils umfangreiche Befehlsgebilde benötigt. Damit sich der Benutzer nicht in jedem Einzelfall wieder mit der Zusammensetzung einer von ihm benötigten Befehlskombination befassen muss, stellt man ihm in einer höheren Programmiersprache¹⁷ ein Befehlsrepertoire zur Verfügung, das der

14 Vgl. als Beispiel: IBM System/38 Technical Developments, IBM S/38 Tech. Dev. 1978 (Hrsg.), S. 19f: «... some functions traditionally found in programming systems are incorporated into the hardware design.» Zur Bedeutung unten Kap. 3 B.

15 ZIMMERMANN W. L., Bd. II, S. 173, definiert das Programm als «eine Aufstellung aller Befehle und der zu verarbeitenden Daten, durch die eine genau umrissene Datenverarbeitungsaufgabe in Einzeloperationen aufgegliedert und deren Ablauf dadurch exakt und eindeutig umschrieben wird».

Vgl. auch DIN 44300: «Eine zur Lösung einer Aufgabe vollständige Anweisung zusammen mit allen erforderlichen Vereinbarungen. Eine Anweisung ist eine in einer beliebigen Sprache abgefasste Arbeitsvorschrift, die im gegebenen Zusammenhang wie auch im Sinne der benutzten Sprache abgeschlossen ist. Eine Vereinbarung ist eine Absprache über in Anweisungen auftretende Sprachelemente.»

Vgl. auch Definitionsvorschlag der CONTU, S. 44: «A <computer program> is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result». Mit dieser Definition sollte nach dem Vorschlag der Kommission in Section 101 beim US-Copyright Act (1976) der Kreis der schützbaren Werke ergänzt werden.

16 Befehl = kleinstes Element einer Verarbeitungsaufgabe. Ein Befehl im Programm entspricht selten einem Hardwarebefehl (feste Folge von Schaltanweisungen). Erforderlich ist ein maschinelles Befehlsauflösungsverfahren mit Hilfe eines Compilers. Ein Befehl wird oft auch als Instruktion oder Statement bezeichnet.

17 Vgl. näheres zu einzelnen Sprachen unter Kap. 2 E III. Nach ZIMMERMANN W. L., Bd. II, S. 225f: «Programmiersprachen sind eindeutig formale Sprachen, mit deren Hilfe der Datenverarbeitungsanlage mitgeteilt wird, in welchen Verarbeitungsschritten eingegebene Daten zu verarbeiten sind.» Die aufgeführten Programmdefinitionen gelten also unabhängig von einer bestimmten Programmiersprache. Gemäss Institut, Bestand Schweiz 1978, Tab. 54, S. 59, verwenden 42,2% der Benutzer COBOL und 37,2%

Denk- und Ausdrucksweise seines Aufgabengebietes möglichst entspricht, d.h. problemorientiert ist. Mit Hilfe eines Übersetzungsprogrammes (Compiler, Umwandlungsprogramm) wird auf dem Computer diese Kunstsprache¹⁸ auf das elementare Befehlsrepertoire der Maschine zurückgeführt. Die zu einem Programm vereinigten Befehle einer höheren, meist symbolischen Programmiersprache werden vor der Übersetzung als Source-Programm bezeichnet. Nach der Umwandlung handelt es sich um das von der Maschine direkt interpretierbare Objekt-Programm (Maschinen-Programm).

Auf dem gleichen Prinzip wie die eben erläuterte Befehlshierarchie beruht die Programmhierarchie: Einige Programme oder Programmteile (Unterprogramme, Moduln, Routinen) finden wiederholt Verwendung für eine Vielzahl von unterschiedlichen EDV-Aufgaben. Bei jeder Verwendung genügt anstelle der jedesmaligen Wiederholung aller Programmschritte ein symbolischer Verweis. Nach dem Kriterium der Verwendungshäufigkeit lässt sich eine – keineswegs zwingende oder exakt abgrenzbare – hierarchische Einstufung vornehmen:

1) *Betriebssystem oder Betriebssoftware*¹⁹

«Ein Betriebssystem kann als hierarchisch aufgebautes Bündel von «Überprogrammen» bezeichnet werden. Ihm obliegt es, den allgemeinen Betriebsablauf zu steuern und das Zusammenwirken aller Komponenten eines Datenverarbeitungssystems zu optimieren²⁰.»

2) *Anwendungssysteme, Anwendersoftware oder Anwendungsprogramme*

Sie dienen der Lösung von konkreten Aufgaben. Zu unterscheiden sind im wesentlichen «massgeschneiderte», anwenderspezifische Pro-

RPG als vorwiegend kommerziell-orientierte Sprachen. Wesentlich bescheidener ist der Anteil der technisch-wissenschaftlich orientierten Sprachen.

Bei den nachfolgenden Detailbetrachtungen und Beispielen werden im besonderen die Möglichkeiten des COBOL zugrundegelegt, wobei dessen Eigenarten ohne weiteres auf andere höhere symbolische Sprachen übertragbar sind.

18 WEICHELBAUMER C., S. 40, unterscheidet natürliche Sprachen (Umgangssprachen, Gruppen- bzw. Fachsprachen) und Formal- oder Kunstsprachen (universelle oder spezielle Sprachen im Bereich der Datenverarbeitung, d.h. hauptsächlichste Hilfsmittel für die Mensch-Maschine-Kommunikation, mathematische Formelsprachen, Notenschrift für Musik usw.).

19 Auch Systemsoftware, Systemprogramme, Organisationsprogramme, Grund- oder systemnahe Software, Software i.e.S. bezeichnet.

20 ZIMMERMANN W. L., Bd. II, S. 256. Die Bezeichnung als «Überprogramme» muss als generelle Überordnung zu den Anwendungsprogrammen verstanden werden. Das «Überprogramm» steuert und kontrolliert deren Ablauf. U.a. werden auch die Übersetzungsprogramme den Betriebssystemprogrammen zugeordnet. Ihre häufig verwendbare Grundfunktion besteht darin, das für die Formulierung von Anwendungsprogrammen verfügbare reichere sprachliche Werkzeug den im allgemeinen bescheideneren Möglichkeiten der Hardware anzupassen.

gramme und Standardprogramme, die mit Berücksichtigung der Aufgabenstellungen eines breiteren Benützerkreises konzipiert wurden. Standardprogramme eignen sich für stereotype Aufgaben wie Rechnungswesen, Materialwirtschaft, Planungsrechnung usw. und können mit wenigen oder gar ohne Änderungen bei einer Vielzahl von Computerbenutzern eingesetzt werden.

Bei den Programmen beider Kategorien und den innerhalb dieser noch weiter unterscheidbaren Programmtypen²¹ handelt es sich um solche im Sinne der obenstehenden Definition²². Sie unterstehen ausnahmslos dem Gebot der Lückenlosigkeit, d.h. alle darin enthaltenen Anweisungen, seien dies Befehle höherer Sprachen oder Verweisungen auf hierarchisch höherstehende Programme oder Programmteile, müssen via Übersetzungsprogramme, Mikroprogramme oder mit Hilfe sonstiger Steuer- und Dienstprogramme der Betriebssoftware in eine lückenlose Folge von Instruktionen des festen Befehlssystems einer bestimmten Datenverarbeitungsanlage auflösbar sein. Obwohl die verschiedenen Programme sich je nach Nähe zur Hardware formell, d.h. in der konkreten sprachlichen Ausgestaltung stark unterscheiden können, kann der Gehalt stets derselbe sein, weil letztlich immer nach einer maschinengerechten Funktionsauflösung für ein bestimmtes Problem gesucht wird. Dabei hat man sich je nach Anzahl und Umfang der verfügbaren softwaremässigen Funktionsmuster (höhere problemorientierte Befehle, Makroinstruktionen, Programm-Moduln etc.) an Befehle zu halten, die einer «festverdrahteten» Funktion der Maschine entsprechen.

3. *Die softwaremässige Kompatibilität*

Wenn eine Symbolik in der Weise übersetzt werden kann, dass für verschiedene Computertypen gleich wirkende lückenlose Funktionsauflösungen entstehen, so liegt eine softwaremässige Kompatibilität vor. Diese Kompatibilität wird heute in aller Regel nur bei Programmen erreicht, die in den verbreitetsten²³ höheren Sprachen geschrieben sind. Mit der Kompatibilität ist eine entscheidende Voraussetzung gegeben, damit vor allem die in höheren Sprachen verfassten Anwendungsprogramme auch ausserhalb des Kreises von Benutzern eines bestimmten Computertyps ohne wesentlichen Anpassungsaufwand verwendungsfähig sind. Aufgrund des eindeutig feststellbaren Vormarsches von einzelnen, auf Computern verschiedenster Bau-

21 Näheres bei ZIMMERMANN W. L., Bd. II, S. 255ff; kurze Übersicht in Anhang Nr. 1.

22 Kap. 2 B II 2 und unten Kap. 2 D.

23 Vgl. oben Kap. 2 B I 2. Allerdings bestehen in der Praxis auch bei höheren Sprachen mit gleicher Bezeichnung oft Inkompatibilitäten.

art einzusetzenden Programmiersprachen²⁴, fallen in zunehmendem Masse kompatibilitätsbedingte Verbreitungshindernisse weg²⁵.

Das Programm wird aus dieser Sicht für einen grösseren Personenkreis begehrenswert, verkehrsfähiger und wohl auch «übernahmegefährdeter».

4. *Das Programm als Bestandteil der Software*

Software und Computerprogramme dürfen nicht, wie das gelegentlich geschieht²⁶, gleichgesetzt werden. Zu eng ist auch die oft durchschimmernde Auffassung²⁷, dass nur der Programmschutz von zentraler Bedeutung ist und die Ergebnisse davorliegender Entwicklungsphasen der Software²⁸ von eher untergeordneter Bedeutung sind. Das Programm stellt zweifelsohne das umfassendste und am unmittelbarsten nutzbare Ergebnis dar; überall dort aber, wo die Übernahme des Programmes für einen Dritten wegen Kompatibilitätshindernissen, fehlender Eignung für eine Hardware-Konfiguration, mangelnder Übereinstimmung mit anwenderspezifischen Lösungserfordernissen oder sonst zu aufwendiger Anpassungsarbeiten²⁹ uninteressant ist, gilt das nicht ohne weiteres auch für die Verwertung von Ergebnissen der Systemanalyse, der Datenfluss- und Programmablaufplanung, weil sich mit deren Übernahme erhebliche Ersparnisse erzielen lassen. Aus diesem Grunde ist die Fragestellung dieser Arbeit betont erweitert auf die konkretisierten Ergebnisse aller Entwicklungsphasen der Software.

24 Vgl. oben Kap. 2 B I 2 und auch Tab. 55, S. 60, der erwähnten Untersuchung (Institut, Bestand Schweiz), die auch innerhalb von Branchengruppen eine starke Konzentration in der Verwendung von einzelnen höheren Sprachen feststellt. Bei Anwendern mit verwandten Aufgabenstellungen erhält die Abwesenheit von Kompatibilitätshindernissen eine besondere Bedeutung.

25 Vgl. auch Feststellungen der CONTU, Final Report, S. 40: «Early programs were designed by machine manufacturers to be used in conjunction with one model or even one individual computer. Today many programs are designed to operate on any number of machines from one or more manufacturer.» Die Kommission macht auf Ähnlichkeiten bei der Entwicklung von Tonträgern aufmerksam (unten Kap. 3 A).

26 Vgl. Hinweis von LUTZ H., GRUR 1976, S. 331; ferner ZIPSE E., GRUR 1973, S. 126, der die Software als «Sammelbegriff für die Maschinenablauf- und Anwenderprogramme» versteht.

27 Vgl. etwa SIDLER P., S. 6. In der Mehrzahl der bisherigen Abhandlungen wird praktisch nur der Schutz des Programmes zur Diskussion gestellt.

28 Unten Kap. 2 C.

29 Die Praxis zeigt, dass die Änderung oder Verbesserung von fremden Programmen recht bald schwierig und aufwendig wird, weil das Programm die detaillierteste Stufe der durch die Menschen zu behandelnden Problemauflösung enthält. Die erforderliche Übersicht zur Einordnung einer erheblichen Änderung in den Kontext lässt sich oft nur nach langem Programmstudium gewinnen. Vgl. weiteres unten Kap. 3 B.

II. *Der dieser Arbeit zugrundegelegte Software-Begriff*

Nach ZIMMERMANN wird der schwer substantiierbare³⁰ Begriff der Software in drei verschiedenen, nicht genau voneinander abgrenzbaren Bedeutungen gebraucht³¹:

(1) «Software (im weitesten Sinne) bezeichnet alle nicht technisch realisierten Objekte (Programme, Modelle, Forschungsergebnisse), die im Zusammenhang mit der elektronischen Datenverarbeitung stehen. Software in diesem Sinne lässt sich als «geistiger Überbau der elektronischen Datenverarbeitung» bezeichnen.»

(2) «Software (im weiteren Sinne) fasst sämtliche Arten von Anwenderprogrammen sowie sämtliche Arten und Ausbaustufen von Betriebssystemen zusammen. Kurz: «Alles was auf Computern läuft und sie zum Laufen bringt».» Die Software in diesem häufigsten gebräuchlichen Sinne wird gelegentlich auch als Programmatur bezeichnet.

(3) «Software (im engeren Sinne) ist der Inbegriff sämtlicher Arten und Ausbaustufen von Betriebssystemen. Mit anderen Worten: «Alles was eine Datenverarbeitungsanlage zum Laufen bringt»,» aber noch nicht unmittelbar zu konkreten Ergebnissen einer Anwendung beiträgt.

Zum vornherein abzulehnen ist der zu enge dritte Begriff, weil er u.a. die wichtige Kategorie der Anwendungsprogramme³² nicht einschliesst. Wie bereits erwähnt³³, ist die im zweiten Begriff enthaltene Beschränkung auf die Programmatur ebenfalls zu eng, weil die in den Programmen enthaltenen Ergebnisse früherer, aber auch diejenigen späterer Entwicklungsphasen nicht erfasst sind, welche nach Hinweisen von zahlreichen Autoren ebenfalls schutzbedürftig sind.

Diese Ergebnisse sind vom weitesten Begriff eindeutig erfasst. Dennoch kann den nachfolgenden Betrachtungen auch dieser Begriff nicht zugrundegelegt werden, weil er eine Anzahl von nicht weiter interessierenden, typisch nachschaffenden Arbeits- und Unternehmerleistungen³⁴ umfasst.

Keiner von diesen in der EDV-Literatur vorgegebenen Softwarebegriffe ist geeignet, alle unter urheberrechtlichem Blickwinkel relevanten Er-

30 Es sei denn, dass man sich mit dem im Grunde nichtssagenden Gegensatz zur Hardware begnügen wollte. Siehe oben Kap. 2 A.

31 Siehe: ZIMMERMANN W. L., Bd. I, S. 77f.

32 Oben Kap. 2 B I 2.

33 Oben Kap. 2 B I 4.

34 Dazu gehören nach OELSCHLÄGER C., S. 10, z.B. Denkvorarbeiten für die Entwicklung, Planung und Schaffung von finanziellen, organisatorischen, personellen, raum-mässigen Voraussetzungen und dergleichen. Ebenso Leistungen des Bedienungs-personals und der Datentypisten, die Software-Ergebnisse (z.B. das codierte Sourceprogramm) auf maschinenlesbare Datenträger übertragen.

scheinungen einzugrenzen. Wie gezeigt, ist aber für die eigene Begriffsbildung am ehesten vom zu weiten ersten Softwarebegriff auszugehen. Zu dessen Einschränkung bietet sich ein dem Urheberrecht entnommenes, selbstverständliches Kriterium an. Mit dessen Beifügung in der zweiten Satzhälfte ergibt sich: Unter Software sind alle im Zusammenhang mit der Datenverarbeitung auftretenden, nicht technisch realisierten Objekte zu verstehen, die «in sinnlich wahrnehmbaren Gebilden in Erscheinung»³⁵ treten bzw. in einem der denkbaren Mitteilungsträger geformt oder konkretisiert sind.

Es muss aber weiter präzisiert werden:

(1) Als Software interessieren nur Objekte, die «zur Lösung eines Datenverarbeitungsproblems erforderlich sind»³⁶, nicht aber solche, die Ergebnis oder Gegenstand eines Datenverarbeitungsprozesses sind. Auszunehmen sind demnach:

– Alle Objekte, bei denen der Computer oder irgendein Automat als Hilfsmittel³⁷ die «Handarbeit» des Gestalters ersetzt oder sogar durch synthetische Aleatorik auf das Werk einen gewissen gestalterischen Einfluss ausübt³⁸. Dies können Schöpfungen der Sprache³⁹, der Musik⁴⁰ oder der Graphik⁴¹ sein.

– Alle Objekte, die im Anschluss an eine datentechnische Aufbereitung als Gegenstand der Datenverarbeitung in Dateien für eine Wiedergewinnung gespeichert sind.⁴²

Die erwähnten beiden Kategorien von Ergebnissen der Datenverarbeitung interessieren uns als Software nur insoweit, als es um den Schutz der diesen Ergebnissen zugrundeliegenden system- und programmtechnischen Lösungsansätze geht. Bei jedem computererzeugten Input (Gegenstand) und

35 ULMER E., Urheberrecht, S. 12.

36 OELSCHLÄGER C., S. 156.

37 Der Umstand, dass der Computer regelmässig zur Unterstützung der Software-Entwicklung eingesetzt wird, findet im folgenden dennoch Beachtung beim Objekt-Programm und den Source-Programmen, die durch den Einsatz von Programm-Generatoren geschaffen werden.

38 Vgl. näheres bei KUMMER M., Werk, S. 170ff; ders., Entgrenzung, S. 107f; FROMM K. F., Der Apparat als geistiger Schöpfer, GRUR 1964, S. 304ff; FABIANI M., Sind Apparate geistige Schöpfer?, GRUR Int. 1965, S. 422ff; MÖHRING Ph., Können technische, insbesondere Computer-Erzeugnisse Werke der Literatur, Musik und Malerei sein?, UFITA 1967, Bd. 50, S. 835ff; SAMSON B., Die Computerkunst und das Urheberrecht, UFITA Bd. 72, 1975, S. 89ff; GIRTH P., S. 13ff.

39 Beispiele bei KUMMER M., S. 184f.

40 Vgl. SCHENK U., S. 128ff; WEISSTHANNER M., S. 87ff; GIRTH P., S. 73ff.

41 Beispiele bei KUMMER M., Werk, S. 170ff.

42 Vgl. näheres u.a. bei GOOSE D., Hindernisse, Sonderdruck, S. 309ff; ders., UFITA Bd. 53, 1975, S. 1-91; ULMER E., GRUR 1971, S. 297ff; ULMER E. und KOLLE G., GRUR Int. 1976, S. 108ff; KATZENBERGER P. und KOLLE G., Informations- und Dokumentationssysteme, Beiträge, S. 182ff; KATZENBERGER P., GRUR 1973, S. 629ff; RINGER A. B., GRUR Int. 1968, S. 18ff; ULMER E., Copyright Problems, in Copyright February 1972, S. 37ff; ders. in Copyright December 1975, S. 240ff; Institut für Film- und Fernsehrecht, in Film und Recht Nr. 5/1976, S. 319ff.

Output (Ergebnis) interessieren uns als Lösung auch gestalterische Elemente, soweit sie als eine mit Rücksicht auf technische und softwaremässige Eigenheiten von Ein- und Ausgabegeräten und dergleichen gefundene Lösung für ein Darstellungsproblem zu betrachten sind. Beispiel: Als Software interessiert bei einem für Buchhaltungsaufgaben eingesetzten Computer mit Ein- und Ausgabe auf Bildschirm nicht die (in Verbindung mit Eingabedaten ermittelte und auf dem Bildschirm gezeigte) konkrete Bilanz, sondern nur die Art und Weise, wie auf einem Bildschirm anstelle des traditionellen Kontoblattes die Ausgabe einer Bilanz darstellerisch ansprechend und zugleich sachlich richtig gelöst werden kann⁴³.

(2) Zu den wahrnehmbaren und damit urheberschutzrelevanten Gebilden der Software gehören auch diejenigen Erscheinungen, bei welchen das vorhandene Ausdrucksmittel für den Menschen nicht jederzeit in unmittelbar wahrnehmbarer Weise vorhanden ist⁴⁴.

(3) Aus dem bisher zugrundegelegten Softwarebegriff ergibt sich, dass die jeder Einzelaufgabe der Software-Entwicklung zugehörige Dokumentation erfasst ist. ZIMMERMANN⁴⁵ führt im Laufe aller Entwicklungsphasen rund vierzig verschiedene Dokumentationen auf, die alle einer urheberrechtlichen Würdigung bedürften.

Aus arbeitsökonomischen Gründen ist deshalb eine Gruppierung nach der Beschaffenheit des Mitteilungsträgers angezeigt (vgl. unten lit. a), zudem sind Auswahlkriterien festzulegen, um den am meisten «übernahmegefährdeten» und deshalb schutzbedürftigsten Ergebnissen eine eingehendere Betrachtung widmen zu können (vgl. unten lit. b).

a) Gruppierung nach der Beschaffenheit des Mitteilungsträgers

Es sind Software-Dokumentationen⁴⁶ zu unterscheiden

– mit vorwiegend umgangssprachlichen Elementen.

– mit zeichnerischen, schematischen (teilweise in Verbindung mit umgangssprachlichen) Elementen.

– mit sprachlichen Elementen, wobei anstelle des ganzen Wortschatzes einer Umgangssprache nur eine beschränkte Anzahl daraus entliehener Wörter oder Satzfragmente im Rahmen einer stark genormten problem- oder maschinenorientierten Kunstsprache verwendet werden.

43 Es geht um den Ein- oder Ausgabe-Layout oder den Anordnungsplan für die Darstellung der den Gegenstand der Datenverarbeitung bildenden Ein- und Ausgabedaten. Vgl. unten Kap. 2 E II 2.

44 Dazu unten Kap. 2 E IV und 6 F I.

45 ZIMMERMANN W. L., Bd. 2, S. 187ff.

46 Vgl. die Betrachtung im einzelnen unten Kap. 2 E.

– in überwiegend maschinenlesbarer Form, deren sprachliche oder numerische Elemente nur mit Hilfe eines Gerätes unmittelbar sinnlich wahrnehmbar gemacht werden können.

b) *Kriterien zur Auswahl der schutzbedürftigsten Software-Bestandteile*

Folgende Gesichtspunkte sind hervorzuheben:

– Verschiedene Dokumentationen sind lediglich vorübergehende Hilfsmittel oder Zwischenergebnisse für die Kommunikation unter den an einem Projekt arbeitenden Spezialisten. Sie haben eher eine provisorische oder projektinterne Funktion und sind später bei Vorliegen des Endproduktes entbehrlich oder nur noch für Wartungsarbeiten erforderlich, d.h. sie brauchen nicht notwendigerweise an den grösseren Personenkreis der Benutzer einer EDV-Applikation verbreitet zu werden. Massnahmen zur Vermeidung einer unberechtigten Verbreitung sind bei diesen etwas einfacher. Gemäss dem ISIS Software Report⁴⁷ erfordert der praktische Einsatz hingegen üblicherweise die Verbreitung folgender Bestandteile an die Benutzer: Objekt-Programme immer, Source-Programme oft nicht oder nur gegen Aufpreis; beinahe immer Systemflussdiagramme, Dateierklärungen, Input-Output-Formulare und Layouts, Operatoranweisungen, Anwender-Handbücher, Fehlerlisten; eher selten Programmablaufpläne, Blockdiagramme, Testdaten u.a.

– Die in den frühen Entwicklungsphasen⁴⁸ anfallenden dokumentierten Software-Ergebnisse haben in der Regel einen bescheidenen unmittelbaren Nutzwert. Sie enthalten zur Hauptsache die in der konkreten Form uninteressanten und urheberrechtlich ohnehin nicht schützbaeren konzeptionellen Ideen, denen erst noch in den arbeitsintensivsten Entwicklungsphasen⁴⁹, namentlich bei der Programmierung, bis ins letzte Lösungsdetail eine qualifizierte Form gegeben werden muss. Zu den Ergebnissen mit hohem unmittelbarem Nutzwert und mit arbeitsintensiven Entwicklungsanteilen gehören u.a. Input-Output-Layouts, Detailablaufpläne, Programme, Anwendungshandbücher⁵⁰.

– Alle Software-Bestandteile können mit einem im Verhältnis zu den Entwicklungskosten zu vernachlässigenden, sehr geringen Aufwand re-

47 ISIS Software Report, Infratest, München 1979; Es handelt sich um den wohl umfassendsten Katalog über die in Deutschland, Österreich und der Schweiz auf dem Markt erhältlichen Software-Produkte. Die hier genannten Bestandteile bilden regelmässig Gegenstand einer Software-Lizenzierung und sind für den Gebrauch und eventuell die Wartung eines Software-Produktes erforderlich.

48 Unten Kap. 2 C.

49 Unten Kap. 2 E III.

50 Zur Auswahl im einzelnen unten Kap. 2 E.

produziert werden. Das gilt sowohl für die in maschinenlesbarer Form vorhandenen Elemente, weil fast jede EDV-Anlage für das Kopieren von Datenträgern eingerichtet ist, als auch für die mit den üblichen reprographischen Verfahren zu vervielfältigenden Dokumentationen. Aus dieser Sicht ergibt sich ein gleichermassen hohes Schutzbedürfnis⁵¹.

Zusammenfassung

Software ist die Gesamtheit der gedanklich erarbeiteten Materialien zur Lösung eines Datenverarbeitungsproblems. Auszuschliessen sind realtechnische Komponenten (Hardware), Gegenstände und Ergebnisse eines Datenverarbeitungsprozesses sowie nachschaffende und unternehmerische Leistungen zur Schaffung von organisatorischen, personellen, raum-mässigen und finanziellen Voraussetzungen für die Software-Entwicklung⁵².

Aus arbeitsökonomischen Gründen wurden Kriterien genannt, die eine gruppenweise und damit bei besonders schutzbedürftigen Softwarebestandteilen eine vertiefte Betrachtung erlauben. Der Konkretisierung des Schutzgegenstandes sind jedoch Grenzen gesetzt, weil kein allgemein anerkannter Dokumentationsstandard existiert und im Einzelfall je nach Projektumfang und -kompliziertheit oder Qualifikation der beteiligten Spezialisten von den bekannten Mustern stark abgewichen wird.

C. Phasen der Software-Entwicklung (Programmierungsphasen)

I. Gegenstand und Verhältnis der Entwicklungsphasen untereinander

Der für eine Problemstellung ermittelte Algorithmus (hier: generelle Problemlösung) und das lauffähige Computerprogramm sind, chronologisch gesehen, erstes und letztes Entwicklungsergebnis⁵³. Zwischen Algorithmus und Programm fallen mehrere schutzrelevante Ergebnisse an, die teils nur eine vorübergehende Hilfsfunktion, teils neben dem Programm eine wichtige begleitende Aufgabe für die Verwirklichung des praktischen Einsatzes haben. Alle diese Entwicklungsergebnisse basieren auf Überlegungen, die den folgenden, im einzelnen nicht genau voneinander abgrenzbaren und im zeitlichen Ablauf nur schwerpunktmässig aufeinanderfolgenden drei Sphären zugehören:

51 Vgl. CONTU, Final Report, S. 39 und S. 41.

52 Im wesentlichen besteht eine Übereinstimmung mit den im WIPO-Vorschlag (vgl. WIPO, Publ. No. 814, Modell, Section I, S. 9) umschriebenen Schutzgegenständen.

53 Zum Ablauf bei der Erstellung von Programmen siehe Anhang Nr. 2.

1. *Sphäre der Aufgaben- oder Problemstellung*

Hierzu gehören Vorgaben und Kenntnisse des Sachgebietes, für dessen Aufgaben der Computer dienstbar gemacht werden will. Das sind bei kommerziellen, administrativen und organisatorischen Anwendungen z.B. Kenntnisse von Buchhaltungsverfahren, revisionstechnischen oder anderen betriebswirtschaftlichen Grundlagen, d.h. Kenntnisse, die aus rechtlicher Sicht zum Gebiet der freien Anweisungen an den menschlichen Geist gehören. Bei wissenschaftlichen Aufgaben handelt es sich z.B. um Kenntnisse von physikalischen oder mathematischen Methoden, bei technischen Aufgaben um Kenntnisse von teilweise patentfähigen Verfahrensregeln.

2. *Systemsphäre*

Dazu gehören Überlegungen, welche die automatengerechte Abgrenzung, Zergliederung, Strukturierung und Umformung der Problemstellung betreffen, u.a. Realisierbarkeitsprüfung, Ermittlung von soft- und hardwaretechnischen Voraussetzungen, Strukturierung von Abläufen. Es geht um die «mit der Lösung einer Datenverarbeitungsaufgabe im ganzen zusammenhängenden Problemkreise ..., wo unabhängig vom jeweils vorliegenden (Anwendungs-)Fall die Gegebenheiten (Elemente) einer Datenverarbeitungsaufgabe und die zwischen diesen bestehenden Beziehungen analysiert werden⁵⁴.»

3. *Programmsphäre*

Dieser Bereich umfasst die mit der Herstellung eines Programmes zusammenhängenden Kenntnisse und Probleme, z.B. Formulierung von Berechnungen oder logischen Verknüpfungen entsprechend dem Befehlsrepertoire und den Regeln einer Programmiersprache, Programmdokumentation zur Vereinfachung des Testverfahrens usw.

Zwischen den Problemen der System- und Programmsphäre, der Domäne der Software-Entwicklung, besteht «eine Hierarchie⁵⁵, die begleitet ist von einem starken gegenseitigen Bezug⁵⁶.» Die bei der Erstellung eines

54 ZIMMERMANN W. L., Bd. 2, S. 181.

55 In der zum Programmschutz vorhandenen juristischen Literatur beinahe ausnahmslos aufgeteilt in Systemanalyse, Datenfluss- und Programmablaufanalyse, Programmdetailplanung oder Blockdiagrammerstellung, Programm-Kodierung; oft auch als die «vier Programmierungsphasen» bezeichnet. Siehe ausführlicher u.a. bei SIDLER P., S. 5ff; KÖHLER R., S. 11ff; ZAHN R., S. 209; SCHWAIGER/KOCKLER, S. 356ff; vgl. Anhang Nr. 2.

56 ZIMMERMANN W. L., Bd. 2, S. 181.

Programmes auftretenden Probleme sind eingebettet in den gesamten Problemkreis der Datenverarbeitungsaufgabe, deren Lösung wiederum durch die bei der Programmerstellung zu erwartenden Probleme stark beeinflusst wird⁵⁷.

II. Die urheberrechtliche Relevanz der Entwicklungs- oder Programmierungsphasen

In der Darstellung der faktischen Grundlagen wird an dieser Stelle Halt gemacht und zu urheberrechtlichen Ausführungen übergegangen, um zu zeigen, dass auf die eingehendere Betrachtung der Entwicklungsphasen verzichtet werden kann. Denn entgegen weitverbreiteter Auffassung ist eine nach Entwicklungsphasen differenzierende Betrachtung für die Beurteilung der Schutzfähigkeit nicht von Bedeutung.

1. Die nach Programmierungsphasen differenzierende Betrachtung

Diese Auffassung geht – kurz zusammengefasst – davon aus, dass mit dem Fortschreiten des Entwicklungsprozesses bis zur Erreichung des Endproduktes «Programm» immer mehr Tätigkeiten und Ergebnisse vorgegeben sind, d.h. sich nach und nach der erforderliche Spielraum für die Leistung eines schöpferischen Beitrages derart stark verengt, dass schliesslich bei den wichtigsten Ergebnissen der letzten Entwicklungsphasen (detailliertes Blockdiagramm, codiertes Quellenprogramm usw.) nur noch von einem routinemässigen, nachschaffenden Vollzug der Ergebnisse der ersten Programmierungsphasen ausgegangen werden kann. Insbesondere wird aufgrund solcher Überlegungen dem in der letzten Phase wirkenden Programmierer ein für die eigenschöpferische Tätigkeit erforderlicher Spielraum abgesprochen, weshalb nur ausnahmsweise – «wenn der vorausgehende Programmablaufplan nur eine grobe Anleitung für die Kodierung enthält, dem Programmierer also ein Spielraum belassen ist⁵⁸» – dem Endprodukt «Programm» ein Urnehmerschutz zugebilligt wird.

Diese Betrachtungsweise, welche die Vielzahl von Wahlmöglichkeiten und die gestalterischen Freiheiten aufgrund der vorangegangenen Programmierungsstufen beurteilt, ist trotz der vorne gezeigten schwerpunktmässig hierarchischen Strukturierung des Entwicklungsvorganges weder im Tatsächlichen zutreffend noch im Rechtlichen erheblich, obwohl SCHWAIGER/

57 Ebd.

58 SCHWAIGER/KOCKLER, S. 358 und dortige Zitate; vgl. auch MOSER H., Vergleichende Literaturuntersuchung, GRUR 1967, S. 639.

KOCKLER⁵⁹ bei allen sonst umstrittenen Einzelfragen als methodische Gemeinsamkeit weitgehend aller Autoren festgestellt haben, dass «regelmässig vier voneinander trennbare Programmierungsphasen» unterschieden und anhand dieser methodisch klar auseinandergehalten werden kann, «inwieweit eine Programmierungsphase durch eine eigenpersönliche, eigenschöpferische und individuelle Tätigkeit gekennzeichnet ist.»

2. *Die Abwendung von einer nach Programmierungsphasen differenzierenden Betrachtung*

a) *Tatsächliches*

Der von den meisten juristischen Autoren festgestellten Vierteilung der Programmentwicklung stehen in der EDV-Literatur zahlreiche andere Phaseinteilungen gegenüber, wo z.B. zehn⁶⁰, sechs⁶¹ oder weniger⁶² Arbeitsgänge unterschieden werden. «Die erwähnte Aufteilung der Programmiertätigkeit in vier Phasen entbehrt nicht einer gewissen Willkürlichkeit und diese Vierphasigkeit dürfte in concreto selten bei einem bestimmten Softwarebestandteil bzw. Programm vorliegen.»⁶³ Sehr unterschiedlich sind die in der Praxis zu beobachtenden, von den schematisierenden Darstellungen der Literatur oft stark abweichenden Vorgehensweisen, wo je nach Umfang und Überblickbarkeit eines Projektes, Routine eines Softwareteams, verfügbaren Hilfsmitteln (Programmgeneratoren, parametrisierbare Programme, computerunterstützte Analyse- und Dokumentationsverfahren usw.) teilweise unter Auslassung von ganzen Phasen verfahren wird. Noch weniger schematisch und phasenweise erfassbar sind die rein geistigen Vorgänge der Programm-Entwicklung, wo gleichzeitig eine Orientierung nach beiden Richtungen der Hierarchie stattfindet. Ferner werden oft – bewusst oder unbewusst – die zu lösenden Probleme zwischen den Phasen verschoben, weil sie entweder erst später erkannt oder für eine Lösung transparent gemacht werden können.

Weiter ist zu beobachten, dass nicht mit jedem Entwicklungsschritt der für eine schöpferische Beitragsleistung erforderliche Spielraum notwen-

59 SCHWAIGER/KOCKLER, S. 356ff, haben die verschiedenen Meinungen zur Schutzfähigkeit der Ergebnisse einzelner Programmierungsphasen zusammengetragen. Vgl. auch KOLLE G., GRUR 1974, S. 9f: Hier wurde beim codierten Programm richtig gesehen, dass letztlich die nach Programmierungsphasen aufgeteilte Betrachtung unerheblich ist, weil die Summe der in den einzelnen Ergebnissen innewohnenden Individualitäten massgebend ist.

60 MARTIN T., Prozessdatenverarbeitung, 1976.

61 ZIMMERMANN W. L., Bd. 2, S. 187ff.

62 PARISINI/WÄCHTER, S. 29ff.

63 SCHWAIGER/KOCKLER, S. 364.

digerweise enger wird, sondern dass sich mit zunehmender Detaillierung immer wieder neue, in ihrer praktischen Auswirkung auf das Endprodukt allerdings bescheidenere Spielräume eröffnen⁶⁴. Die Bedingungen der Problemlösung verdichten sich, ohne dass parallel dazu ein wesentlicher Verlust an ausfüllbaren Spielräumen festzustellen ist. Individuelle Elemente im kleinen und im grossen sind aus urheberrechtlicher Sicht, soweit im Bereich des «Handgreiflichen», gleichzustellen. Individuelle Beiträge im Konzept und in den vergleichsweise unbedeutenderen Details eines Programmes haben in gleichem Masse die Eigenschaft, den Urheberschutz begründen zu können.

Die Software-Entwicklung spielt sich demnach – zumindest inhaltlich⁶⁵ – nur bedingt in einem thematisch und für die Lokalisierung von schöpferischen Elementen klar strukturierbaren Schöpfungsprozess ab. Wegen der zu Beginn der Entwicklung vorhandenen Rahmenbedingungen, bestimmt durch das Problem und die verfügbaren Lösungsmittel (z.B. Befehlsrepertoire), muss von einer deutlichen Wechselbeziehung und Verflechtung der einzelnen Tätigkeiten ausgegangen werden. Aufgrund von Gliederungen des Entwicklungsvorganges lassen sich weder einzelne Beiträge lokalisieren und schöpferisch bewerten noch Ansätze einer kategorialen Methode für die Zuordnung der Urheberschaft⁶⁶ begründen. Anders als bei einem Filmwerk⁶⁷, «wo die einzelnen Werkbeiträge ... entsprechend ihrer Funktion charakteristische und regelmässig gleichbleibende Merkmale aufweisen⁶⁸» und jeder Tätigkeit einen typischen Bezirk von schöpferischen Qualitäten zugewiesen werden kann, gilt bei der Software, dass wegen des ausgeprägt lösungsbestimmten wechselseitigen Bezuges die einzelnen Werkbeiträge funktional austauschbare Merkmale haben und sich die schöpferischen Bezirke verschieben lassen. Das symbolische Programm kann – und wird oft – ohne Programmablaufplan oder sonstige weitergehende konzeptionelle Vorlagen geschrieben, bei seiner Niederschrift können Fehler und Lücken des Konzeptes erkannt und richtiggestellt werden⁶⁹ usw. In diesen Fällen können

64 Zutreffend ULMER E., Wiss. Werke, S. 17.

65 Als inhaltliche Elemente werden im folgenden Algorithmus, Problemlösung, Lösungsidee, Lösungsanweisungen und dergleichen synonym behandelt. Gemeint sind Erkenntnisse oder Ideen zur Lösung eines EDV-Problems, die als Erkenntnisgegenstand der Software zum Ausdruck gelangen. Vgl. unten Kap. 2 D.

66 Die nach Programmierungsphasen getrennte Untersuchung der Tätigkeiten erfolgte bei allen Autoren im Hinblick auf die Frage nach dem urheberrechtlichen Schutz der Programme und nicht – wie der Denkansatz eher annehmen liesse – zur Klärung des Verhältnisses der an der Softwareherstellung Beteiligten (Miturheberschaft, Bearbeitung, Werkverbindung oder Sammelwerk).

67 Ein Vergleich, der zuerst von ULMER E., Wiss. Werke, S. 17, angestellt wurde, weil bei der Software- und Filmherstellung in ähnlicher Weise stufenweise und arbeitsteilig auf ein Endprodukt hingearbeitet wird.

68 BOHR K., Die Urheberrechtsbeziehungen der an der Filmherstellung Beteiligten, Diss. Saarbrücken 1978, S. 41.

69 Beim Film können z.B. der Kameramann oder der Maskenbildner im Rahmen ihrer Funktion nicht wesentliche schöpferische Elemente beibringen, die in den Aufgaben-

sich in der einen letzten Tätigkeit (Werk letzter Hand) alle schöpferischen Elemente kumulieren, die nach der üblichen Auffassung in früheren Programmphasen auftreten müssten.

Im Unterschied zu den oben besprochenen inhaltlich schöpferischen Elementen bewahrt das Ausdrucksmittel der verschiedenen Beiträge die seiner Funktion entsprechenden charakteristischen Merkmale, was aber nicht heisst, dass die verschiedenen Phasen im Ausdrucksmittel nicht voneinander abhängig sind. Der Programmablaufplan besteht immer aus darstellerischen Elementen, das Konzept und das Programm aus sprachlichen. Die in einer typischen Form verarbeiteten inhaltlichen Elemente können sich aber – wie soeben festgestellt – stark verschieben.

b) *Rechtliches*

Sowenig wie die Einteilung in Programmierungsphasen im allgemeinen mit den tatsächlichen Gegebenheiten übereinstimmt, darf sie auch für die rechtliche Beurteilung als erheblich angesehen werden. Wenn stufenweise untersucht wird, «ob und inwieweit für die einzelnen Programmierungsphasen ein urheberrechtlicher Schutz in Betracht kommt⁷⁰», werden zwei Probleme ineinander verwoben: Einerseits die primär interessierende Frage der Urheberschutzfähigkeit, der Werkqualität der verschiedenen Zwischenprodukte und des Endproduktes «Programm» und andererseits die sekundäre Frage der Beschaffenheit und gegenseitigen Abhängigkeit von einzelnen schöpferischen Beiträgen als Kriterium für die Zuordnung des Urheberrechts an die an der Software-Entwicklung beteiligten Personen. Zu entscheiden ist nach dem heutigen Stand der Diskussion noch immer primär über die Art der Werke und über das Ob des Urheberschutzes und nicht über die Zahl der Werke und über das Wie des Schutzes⁷¹. Zunächst muss die Beschaffenheit der schützbaren Gebilde feststehen, bevor die mehr technischen Rechtsfolgen, ob und welcher Beitrag einer anderen Werkstufe eine eigenschöpferische Bearbeitung, freie und unfreie Benützung, abhängige Neuschöpfung usw. ist, entschieden werden können. Das Programm (wie auch andere Softwarebestandteile) verbindet als einheitliches Ganzes die

kreis des Filmkomponisten, Filmarchitekten oder Autors des literarischen Stoffes gehören. Die einzelnen Beiträge können sich allenfalls etwas kompensieren, es bleibt aber z.B. der spezifische Beitrag des Kameramannes, wenn er durch originelle Bildeinstellungen einen dürtigen literarischen Stoff kurzweilig gestalten kann. Anders bei der Software. Der Programmierer kann ohne weiteres die Beiträge des Organisators, des Analytikers usw. in seinen Beitrag integrieren.

70 Vgl. die Zusammenstellung bei SCHWAIGER/KOCKLER, S. 358.

71 Vgl. zu dieser Unterscheidung eingehender STROH R., *Werkeinheit und Werkmehrheit im Urheberrecht*, Diss. München 1969, S. 41ff.

Elemente (Teilergebnisse) des Gesamtvorganges (Vorstufen). Sämtliche Phasen des Entwicklungsvorganges zielen auf die Vollendung eines bestimmten Endproduktes. Es bildet schliesslich eine funktionelle (verwertungsfähige) und begriffliche Einheit, obwohl es im Einzelfall das Ergebnis der Vereinigung oder der Zusammenstellung mehrerer verschiedener und schöpferischer Werke oder Tätigkeitsarten sein kann. Auch wenn aufgrund einer derartigen Konstellation kein einheitliches, umfassendes Urheberrecht zuerkannt werden kann, genügt für die objektive Schutzqualifikation des Programmes (oder anderer gestufter Werkteile), dass insgesamt der schöpferische und spezifische Charakter gegeben ist. Die materiellen Schutzvoraussetzungen sind erfüllt. Ob der Schutz in Form eines einzelnen oder mehrerer Urheberrechte, für Elemente eines Werkes oder mehrere Werke gewährt werden kann, mag als zweitrangige Frage dahingestellt bleiben. Bei der Beurteilung des Programmes interessiert also nicht in erster Linie, inwieweit die Tätigkeit des Programmierers (Kodierers) im Verhältnis zu den Ergebnissen früherer Phasen schöpferisch ist oder ob sie, wie oft angenommen, nur routinemässige Umgestaltung oder minimale schöpferische Bearbeitung eines vorgegebenen Konzeptes (Original- oder Vorlagewerkes) ist, sondern in welchem Umfange im konkreten Programm insgesamt schöpferische Elemente aller Vorstufen vereinigt sind. Dazu stellt KOLLE⁷² mit Recht fest, dass die Gesamtheit oder zumindest die Mehrheit der aus früheren Tätigkeiten (Problemanalyse, Aufzeichnung von Flussdiagrammen usw.) resultierenden Individualitäten «dem Programm stets innewohnen». Dieser Grundsatz ist ohne weiteres übertragbar auf das Verhältnis von anderen Entwicklungsergebnissen früherer Phasen.

Das Programm geniesst also beispielsweise den Schutz, wenn eine vom Analytiker vorgezeichnete originelle Druck- oder Bildschirmdarstellung (als Bestandteil einer früheren Programmierungsphase) vom Programmierer in mechanischer, nach den Regeln einer Programmierungssprache verlaufender Definitionsarbeit in die Sprachform dieses Programmes übernommen wird⁷³. In diesem Fall kann nicht der gesonderte Beitrag des Programmierers den Schutz begründen, weil eine gegebenenfalls unschöpferische Übernahme oder Nachbildung eines vorbestehenden Werkes vorliegt. Der Schutz liesse sich nur daraus ableiten, dass im Programm in umgestalteter Form geschützte und ungeschützte individuelle Elemente eines Vorlagewerkes enthalten sind, die bei einer Vervielfältigung des Programmes ebenfalls verbreitet werden müssten.

72 KOLLE G., GRUR 1974, S. 9 und dort zit. Literatur.

73 Unten Kap. 6 B II. Es liegt auf der Hand, dass im Programm die vom Analytiker skizzierte Darstellungsform exakt definiert werden muss, damit der Computer in der Lage ist, diese Darstellung nachzuvollziehen.

Wie bei einem Filmwerk werden im Programm Beiträge verschiedener Werkstufen zu einem Gesamtwerk aufgeschichtet. Urheberrechtlich geschützte Befugnisse können sich aus der Abhängigkeit von vorbestehenden Werken, verbundenen Werkteilen oder aufgrund des Zusammenwirkens von Miturhebern ableiten lassen. Dieser Vergleich wäre auch zulässig, wenn die Meinung vertreten werden sollte, dass es sich beim Programmieren nur um eine freie Benützung von ungeschützten schöpferischen Elementen (weil wissenschaftlichen bzw. verstandeswerklichen Charakters) der Vorstufen handle, die den Schutz der abhängigen Gestaltung nicht mehr begründen könne. Wie noch zu zeigen sein wird⁷⁴, gehen beim Programmieren aus den Vorstufen nicht nur ungeschützte schöpferische, sondern auch eine Anzahl von geschützten schöpferischen Elementen der Form ein (umgangssprachliche Textteile, Druck-, Bildschirm- und Dateidarstellungen, Teile der Symbolik usw.) sowie, nach einem Teil der Lehre, (allerdings in sehr bescheidenem Umfang geschützte) inhaltliche Elemente (inhaltliches «Gewebe» vorbestehender Werke). Ausgeschlossen ist aber auch hier nicht, dass auf der Ebene der Programmierung selber noch individuelle Elemente hinzukommen können.

Ergebnis

Für die Fragestellung dieser Arbeit ist es unerheblich, in welcher Programmierungsphase ein Ergebnis geschaffen wurde und inwieweit dieses Ergebnis in bestimmten Etappen individuell geprägt werden konnte. Für die Beurteilung der materiellen Schutzvoraussetzungen ist allein entscheidend, welche Gesamtheit an individuellen Elementen in einem geformten einheitlichen Ergebnis vereinigt ist. Methodisch kann in den nachfolgenden Untersuchungen in der Weise vorgegangen werden, dass anhand eines Kataloges von Softwareergebnissen alle jeweils darin vorkommenden Elemente auf ihre individuelle Beschaffenheit hin untersucht werden. Ein anderes Vorgehen würde mit Blick auf die stark abweichenden Schaffensvorgänge keine allgemeingültigen Aussagen zulassen.

D. Der Erkenntnisgegenstand von Algorithmus und Programm

Mit den nachstehenden Ausführungen soll gezeigt werden, welche Anforderungen an einen Algorithmus und an das Endprodukt «Programm» gestellt werden. Dies als Orientierungsrahmen für die Unterscheidung der

inhaltlichen Elemente von formalen Bestandteilen einer automatisierten Problemlösung. Es ist daraus zu entnehmen, nach welchen Verfahrenselementen bei der Softwareentwicklung gesucht wird, was also Erkenntnisgegenstand einer automatenberechtigten Problemlösung ist.

Die theoretischen Grundlagen der Programmierung werden von der sogenannten Algorithmentheorie untersucht. Wie ZIMMERMANN⁷⁵ zeigt, kann von der «grundsätzlichen Gleichheit» von Algorithmus und Programm ausgegangen werden, weil die aus dem Algorithmusbegriff zu entnehmenden «sieben grundlegenden Bedingungen auch im Programm erfüllt sein müssen».

Im einzelnen entsprechen sich die Bedingungen wie folgt⁷⁶:

I. Bedingungen eines Algorithmus

Ein eindeutig definiertes System von Eingangsgrößen (1.) muss in einer eindeutig beschriebenen (2.) Folge von Arbeitsschritten (3.) umgeformt werden können, wobei die für die Umformung zulässigen Operationen so anzugeben (4.) sind, dass sie automatenverständlich (5.) sind und das zu erzeugende eindeutige System von Ausgangsgrößen (6.) in einer endlichen Zahl von Arbeitsgängen (7.) erreicht werden kann.

II. Anwendung dieser Bedingungen auf das Programm

Ein eindeutig definiertes System von zu verarbeitenden Daten (samt Beschreibung ihrer Rahmenbedingungen) (1.) muss in einer eindeutig beschriebenen (2.) Reihenfolge von Befehlsschritten (3.) umgeformt werden können, wobei die für die Umformung verwendbaren Befehle derweise in einer Kunstsprache (Programmiersprache) anzugeben (4.) sind, dass sie entweder direkt maschinenverständlich sind oder nach bestimmten Regeln maschinenverständlich gemacht (5.) werden können (z.B. mit Hilfe der Betriebssoftware) und das zu erzeugende eindeutige System von Verarbeitungsergebnissen (6.) in einer endlichen Zahl von Arbeitsschritten (7.) erreicht werden kann.

III. Zusammenfassung

Zu den inhaltlichen Elementen der Software gehören demnach Kenntnisse über:

75 ZIMMERMANN W. L., Bd. 2, S. 173ff.

76 Die einzelnen Bedingungen sind numeriert (1.), (2.), usw.

1. Vorhandensein und Eigenschaften von Grössen, die mit Hilfe eines Automaten umgeformt werden können.

2. Die eindeutige logische Ordnung und Reihenfolge des Ablaufs von automatisierbaren Umformungsprozessen.

3. Die Eigenschaften der erforderlichen Umformungsprozesse (Operationen), um aus bestimmten Eingangsgrössen bestimmte Ausgangsgrössen zu erzeugen.

4. Die Methode zur direkten oder mittelbaren Anpassung der erforderlichen Umformungsprozesse an die technischen Gegebenheiten eines Automaten, d.h. Kenntnisse über Modalitäten des Gebrauchs (z.B. verfügbares Repertoire an Umformungsmöglichkeiten) und der Wirkungsweise eines Automaten.

5. Vorhandensein und Eigenschaften von Grössen, die mit Hilfe des Automaten aus bestimmten Ausgangsgrössen in einer endlichen Zahl von Arbeitsgängen erarbeitet werden können.

E. Katalog der wichtigsten Ergebnisse der Softwareherstellung

Die nachstehende Aufzählung folgt den Auswahlkriterien, die oben in Kap. 2 B II 3 b festgelegt wurden. Die Gliederung folgt der oben in Kap. 2 B II 3 a erwähnten Unterscheidbarkeit von verschiedenen vorherrschenden Ausdrucksmitteln. Vorerst geht es nur darum, das Inventar samt den zugehörigen allgemeinen faktischen Grundlagen auszubreiten. Die Betrachtung und Würdigung von urheberrechts-relevanten Charakteristiken folgt in Kap. 6 A bis 6 D.

I. Umgangssprachlich geformte Ergebnisse, Software-Dokumentationen

Die Umgangssprache ist das wichtigste Ausdrucksmittel für die Systemplanung oder Analyse und für die praktische Anleitung der Softwareanwender. Es handelt sich um Texte mit ergänzenden zeichnerischen Darstellungen, die den Inhalt oder den Gebrauch eines Softwaresystems beschreiben.

1. Allgemeine Systembeschreibung

Die Systembeschreibung vermittelt einen generellen Überblick über die Art der gelösten Probleme, über wichtigste Ergebnisse, Lösungstechniken und Rahmenbedingungen für den Einsatz der Software.

2. *(Umgangssprachliche) Programmbeschreibung*

Die Programmbeschreibung enthält eine verbale Schilderung aller Aufgaben und Bedingungen, die beim Kodieren des Programmes zu berücksichtigen sind.

3. *Datei- und Datenbeschreibungen*

In den Datei- und Datenbeschreibungen sind neben technischen Angaben über Zugriffsart, Speichermedium usw. sämtliche Informationen über Datensätze und Datenfelder der Ein- und Ausgabe enthalten, u.a. Aussage und Eigenschaften eines Datenelementes, Mutationsbedingungen, Umformungsregeln.

4. *Bedienungsanleitungen, Anwendungshandbücher (Manuals)*

Bedienungsanleitungen und Handbücher enthalten Anweisungen und Erklärungen für die praktische Handhabung der Software auf allen Anwenderebenen (Operateur, Sachbearbeiter der Datenvorbereitung und Datenverwertung). In diese Gruppe gehören ebenfalls Handbücher, die Eigenschaften von Computersprachen und Betriebssystemen als Voraussetzung für deren Anwendung erklären.

5. *Weitere Ergebnisse dieser Gruppe*

Erwähnt werden in der EDV-Literatur⁷⁷ u.a. noch: Abkürzungsverzeichnisse, Glossare, Formelsammlungen, Fehlerverzeichnisse, Anleitungen für Fehlerbehebung, Erläuterungen des Programmierers im Source-Programm.

II. *Ergebnisse mit zeichnerischen, schematischen oder tabellarischen Darstellungen*

Diese Darstellungen treten vor allem in der System- und Programmplanung, gelegentlich auch in der Softwaredokumentation (oben Abs. I) auf. Elemente solcher Darstellungen sind Linien und auf Konvention beruhende Zeichensymbole⁷⁸ für Speichermedien oder Verarbeitungsfunktio-

77 Siehe PARISINI/WÄCHTER, S. 121ff (Überblick), S. 221–265, Auszug mit Beispielen; ZIMMERMANN W. L., Bd. 2, S. 187ff; DANIELS/YEATES/ERBACH, S. 113ff.

78 Vgl. etwa DIN 66001.

nen sowie Ziffern- und Buchstabenanordnungen. Meistens sind die darstellerischen Elemente kombiniert mit umgangssprachlichen, z.B. mit Einzelwörtern, Wortkürzungen, Satzfragmenten.

1. *Flussdiagramme verschiedener Ebenen*⁷⁹

Die Flussdiagramme dienen der Veranschaulichung von Datenflüssen und Funktionsabläufen. Je nach Entwicklungsebene und Gegenstand werden unterschieden: Diagramme der Systemfunktionen, die den Informationsfluss (Ein- und Ausgabe) unabhängig von der Computertätigkeit darstellen; Systemflussdiagramme, welche den Fluss der Dokumente rund um den Computer darstellen und die in die EDV-Anwendung einzubeziehenden Funktionen und Verfahren von nicht automatisierten Vorgängen separieren; Computerflussdiagramme oder Datenflusspläne, welche den Ablauf der Programme aufzeigen und die davon berührten Dateien bezeichnen; schliesslich die umfangreichsten Diagramme des Programmablaufs, auch Blockdiagramm oder Programmablaufplan genannt, welche die Verfahrensweisen innerhalb eines Programmes bzw. innerhalb der Zentraleinheit separat oder in Ergänzung zu einer verbalen Programmbeschreibung definieren. Das Programmablaufdiagramm ist neben der in der Praxis oft vorgezogenen verbalen Programmbeschreibung entbehrlich. Entgegen ZIMMERMANN⁸⁰, der die Programmabläufe als die beste und verbreitetste Methode der Programmbeschreibung bezeichnet, werden diese in der Praxis je nach Komplexität und Umfang des Projektes nur teilweise oder in geringer Detaillierung als vorübergehendes Hilfsmittel für die Programmkodierung angewendet. Ihre Bedeutung ist sehr gering.

2. *Layouts von Ein- und Ausgabemedien*

Im Layout wird die äussere Form oder die Anordnung definiert, nach der ein Computer die Daten von einem Eingabegerät entgegenzunehmen oder auf ein Ausgabegerät auszugeben hat⁸¹. Bei Datenmedien, die

79 Nach DIN 66001 als Datenflusspläne und Programmabläufe bezeichnet. Die Terminologie ist in der Literatur unterschiedlich, ebenso der damit bezeichnete Darstellungsgegenstand. Mischvarianten sind in der Praxis häufig. Das Darstellungsprinzip ist in allen Fällen ähnlich. Die folgende Darstellung richtet sich nach der Terminologie von DANIELS/YEATES/ERBACH, S. 116. Beispiele im Anhang Nr. 3 und 4.

80 ZIMMERMANN W. L., Bd. 2, S. 201.

81 Diese Layouts (vgl. Beispiele im Anhang Nr. 5 und 6) sind darstellerisches Beiwerk der mit dem Computer vorgesehenen Realisation, dessen In- und Output. Die feste Anordnung dieser Layouts bzw. die mit dem Programm an ihnen ausgeführten Anpassungen werden vom Schöpfer vorgedacht und stehen unter dessen ursprünglicher individueller Einflussnahme. Die Individualität jeder Konkretisierung (der Doku-

nicht für die unmittelbare Wahrnehmung durch den Menschen bestimmt sind (Daten auf magnetischen Datenträgern, Lochkarten usw.) besteht der Layout aus einer Aufzählung und Beschreibung von einzelnen Datenelementen. Bei den für die unmittelbare menschliche Wahrnehmung bestimmten Medien, z.B. Druck- oder Bildschirmdarstellungen, kommen zu der blossen Aufzählung und Beschreibung von Datenelementen noch darstellerische Dimensionen der hierfür verwendeten Geräte in räumlicher, sprachlicher und zeitlicher Hinsicht hinzu. Die vom Analytiker gewählte Datenanordnung wird dem Programmierer für die Definition im Programm in einer auf Konvention beruhenden Ziffern- oder Buchstabenanordnung angezeigt.

3. *Weitere Ergebnisse dieser Gruppe*

In der Literatur werden zahlreiche weitere, dieser Gruppe zugehörige Ergebnisse aufgeführt: Entscheidungstabellen, Speicherbelegungspläne, Zeitpläne, Programmtestpläne, Testdatenverzeichnisse, Systempläne, Laufzeitpläne, Testdatenausdrucke, Bibliotheksverzeichnisse usw.

III. *Sprachlich geformte Ergebnisse mit Verwendung einer genormten maschinen-, symbol-, problem- oder generator-orientierten Kunstsprache*

Die Programmierung oder Kodierung, d.h. die Umwandlung einer allgemein definierten Problemlösung in eine maschinell verwertbare Sprachform, ist ein sehr aufwendiger und wegen der Vielzahl formaler Regeln fehleranfälliger Vorgang. In der Literatur wird das Auffinden des Lösungskonzeptes oft als Hauptsache bezeichnet⁸² und das Schreiben des Programmes zu einer «keine grossen Schwierigkeiten» bietenden «Routinearbeit» verniedlicht. Jedoch ist, zumindest vom Aufwand her gesehen, die mit viel Kleinarbeit verbundene qualifizierte «Formgebung»⁸³ von erheblicher Be-

mentationsskizze und ihre Realisationen mit dem Computer) fliesst allein aus der Persönlichkeit des Urhebers. Es kann deshalb ein Werk vorliegen. Die Hinzufügung von aleatorischen Effekten des Computers ist für die Zwecke der reinen Darstellung von den hier im Vordergrund stehenden Ergebnissen der kommerziellen und technisch-wissenschaftlichen EDV eine Ausnahme. Die entstehungsmässige Verbindung des Werkes zum Menschen ist bei diesen Erzeugnissen bis und mit deren Realisation auf dem Computer als Hilfsgerät in der Hand des Menschen gewahrt und bietet deshalb keine besonderen urheberrechtlichen Probleme (GIRTH P., S. 110; KUMMER M., Werk, S. 170ff: Ursprüngliche menschliche Individualität mit technischer Realisation durch Automaten ohne Aleatorik; HEIM E., Statist. Einmaligkeit, S. 49 und dortige Zitate).

82 ZAHN R., S. 208 mit Hinweisen; SCHWAIGER/KOCKLER, S. 363.

83 CONTU, S. 37: «Computer programs are prepared by a careful fixation of words, phrases, numbers and other symbols in various media.» Hervorgehoben wird hier, was zur eigentlichen aufwendigen Formgebung gehört.

deutung. Dementsprechend gross dürften auch die Schutzinteressen sein. Wenn auch Verallgemeinerungen und genaue Abgrenzungen bei der Aufwandverteilung in der Softwareherstellung schwierig sind, dürfte die Aussage von GALBI⁸⁴ in der Tendenz zutreffend sein. Er meint: «... Between five and twenty percent of investment goes towards developing new concepts and ... in the neighborhood of eighty to ninefive percent goes into the hard work involved in writing code, testing, de-bugging and documenting⁸⁵.» «Die in der Literatur im allgemeinen zum Ausdruck gebrachte Überbewertung des urheberrechtlich nicht schützbaeren Konzeptes, der Lösungsidee, steht in einem Gegensatz zur wirklichen Bedeutung, die der Ausdrucksform, dem Text eines Programmes zukommt»⁸⁶.

Es überrascht deshalb nicht, dass grosse Anstrengungen darauf verwendet werden, die Programmerstellung durch Computerunterstützung zu rationalisieren, wobei «in der vorläufig letzten Stufe versucht wird, dem EDV-Anwender die Notwendigkeit der Transformierung einer zu lösenden Aufgabe in eine höhere Programmiersprache zu ersparen»⁸⁷. Daraus folgt aber auch, dass nicht verallgemeinernd vom Schutz der Computerprogramme gesprochen werden darf, sondern je nach Umfang der Computerunterstützung die Formalismen und Ausdrucksmöglichkeiten anders zu beurteilen sind:

1. *Maschinensprachen und maschinenorientierte Assemblersprachen*

Die Handhabung von maschinenorientierten Sprachen ist aufwendig, umständlich und deshalb immer weniger gebräuchlich. Sie weisen bei weitem nicht die einer Symbolsprache innewohnenden zahlreichen Ausdrucksmöglichkeiten auf. Jedes Verarbeitungsproblem muss bei ihnen mehr oder weniger direkt auf das elementare Repertoire von Maschinenoperationen «reduziert» werden. Wir haben uns im folgenden nur soweit mit ihnen zu befassen, als der Schutz der von der EDV-Anlage aufgrund von höheren Sprachprogrammen selbst erstellten Maschinenprogramme (Objektprogramme) in Frage steht.

84 GALBI E., Proposal, S. 282.

85 Diese Aussage trifft nicht genau die eben genannte Unterscheidung. Es lässt sich aber daraus folgern, dass nicht die Entwicklung des Konzeptes oder der Lösung, sondern der für die EDV-Realisierung erforderliche Vorgang der Formgebung, darunter die bedeutende Kleinarbeit bei der Programmkodierung und Fehlerbehebung, den Hauptteil des Aufwandes verursacht.

86 So zu recht MOOERS C. N., ACM Comp. Surveys, S. 47.

87 ZIMMERMANN W. L., Bd. 2, S. 212.

2. *Höhere symbolische Programmiersprachen*

Wichtig sind heute die «dem menschlichen Verständnis⁸⁸» und damit der menschlichen Verständigungsform angenäherten höheren symbolischen Programmiersprachen. Hierbei stehen die vor allem bei Anwenderprogrammen verwendeten höheren oder problemorientierten Sprachen im Vordergrund⁸⁹. Charakteristisch ist bei ihnen, dass die durchzuführenden Operationen in (meist der englischen Umgangssprache entnommenen) normierten Satzfragmenten, Wörtern oder mnemotechnisch vorteilhaften Symbolen angegeben werden. Operanden (das sind Merkmale für Datenelemente, Datensätze, Dateien, Verzweigungsadressen und Unterprogramme) werden mit frei wählbaren, meist leicht zu merkenden Buchstaben- oder Ziffernkombinationen sowie dem allgemeinen Wortschatz entnommenen Begriffen bezeichnet. In diesen Programmen können in beliebigem Umfang Bearbeitungshinweise, Programmerklärungen, Gliederungshinweise oder sonst dem Verständnis des Programmes förderliche Dokumentationen in frei wählbaren umgangssprachlichen Formulierungen oder Darstellungen integriert werden. Unsere nachfolgenden urheberrechtlichen Betrachtungen beziehen sich vor allem auf diesen häufigsten Programmtypus. Diesem Typus entsprechen meistens die direkt geschriebenen Programme. Die Programme können auch maschinell aufgrund von höheren Norm-, Struktur- oder Generatorsprachen auf diesen Typus zurückgeführt werden.

3. *Norm- und Struktursprachen*

Norm- und Struktursprachen unterscheiden sich gegenüber den soeben erwähnten dadurch, dass Freiheiten im Programmaufbau und in der Symbolik eingeschränkt sind. Anhand von sehr problemspezifischen Verarbeitungsanweisungen werden von der EDV-Anlage vollständige oder für die besondere Ergänzung vorbereitete Anwenderprogramme in einer symbolischen Programmiersprache geschaffen. Es stellen sich hier Fragen des Schutzes von Parameterangaben und, wie übrigens auch bei Ziff. 2 oben, des Verhältnisses vom Norm-, Struktur- und Umsetzungsprogramm zum generierten Programm.

4. *Generatorsprachen*

Das Prinzip ist ähnlich wie bei den soeben besprochenen Sprachen. Generatorsprachen bieten Vorteile der grösstmöglichen Problemnähe. Der

88 ZIMMERMANN W. L., Bd. 2, S. 255.
89 Oben Kap. 2 B I 2; Beispiel in Anhang Nr. 7.

Weg von der allgemeinen Lösung zum Programm wird abgekürzt, indem der Generator im Rahmen von in der Grundstruktur verwandten Problemstellungen automatisch das für den Einzelfall gewünschte Programm erzeugt. Das Generatorprogramm wird mit Verarbeitungsanweisungen gesteuert, die direkt auf die in der Analysephase bereitgestellten formal gebundenen Beschreibungen der allgemeinen Problemlösung zurückgehen, z.B. in Form von Entscheidungstabellen. Mit Blick auf Ziff. 3 oben ergeben sich wenig neue Gesichtspunkte.

5. *Andere Sprachen*

Auf andere Sprachen wie Makrosprachen, Job-Control-Sprachen, NC-Sprachen, Datenbanksprachen usw. wird nicht näher eingetreten, weil es entweder nur terminologische Unterschiede gibt oder sie zu speziell sind.

6. *Programm-Standards, Programmrichtlinien*

Anders als es aus urheberrechtlicher Sicht erwünscht ist, besteht in der EDV-Praxis das Bestreben, möglichst alle vom Programmierer frei gestaltbaren Bereiche zu reglementieren, damit die Programme für Dritte vergleichbarer, schneller überblickbar und verständlicher sind. Durch derartige Massnahmen kann auch die Programmwartung rationalisiert werden. Richtlinien werden hauptsächlich im Bereich der Merkmalsbildung, der Programmstrukturierung, der Auswahl von alternativen Ablaufvarianten und der Programmdokumentation vorgegeben. Diese zusätzlich bestimmten Richtlinien sind in den nachstehenden Betrachtungen ähnlich zu würdigen wie die einer Programmiersprache direkt zugehörigen Regeln.

7. *Source- und Objektprogramme und deren Formen*

Der vom Programmierer geschriebene oder von einem Generator erzeugte Sourcecode wird mit Hilfe des für die gewählte Sprache vorhandenen Umsetzungsprogrammes oder Compilers⁹⁰ in einen Objektcode umgewandelt. Beim Sourcecode sind folgende Erscheinungsformen zu unterscheiden: Das Programm auf Programmblättern (Kodierblättern); nach der Datenerfassung das Programm auf einem maschinenlesbaren Datenträger (Kartendeck, magnetische Datenträger), der für die Eingabe in das Kompilationsverfahren dient; nach der Kompilation das Programm auf einem

⁹⁰ Dazu terminologisch DIN 44300: Anstelle Umsetzungsprogramm oder Compiler auch Kompilierer oder Assembler.

Ausdruck mit Diagnosemeldungen⁹¹. Das Objektprogramm selber existiert meistens nur auf einem magnetischen Datenträger, ab diesem wird es bei jeder Programmdurchführung in den Zentralspeicher kopiert⁹².

8. *Ausblick*

Mit zunehmendem Fortschritt der Compiler- und Generatortechnik verschiebt sich häufiger ein heute noch klar abgrenzbarer Programmierungsvorgang in die Analysephase. Die in dieser Phase vorhandenen freien Darstellungsformen (umgangssprachliche Beschreibungen usw.) unterstehen in vermehrtem Masse formalen Regeln, damit sie direkt als Parameterangaben in einen programmerzeugenden maschinellen Ablauf eingebracht werden können. Daraus folgt, dass das Problem des Schutzes von Parameterangaben an Bedeutung gewinnt. Anzunehmen ist auch, dass die urheberrechtliche Beurteilung der Analysedokumentationen künftig von etwas anderen Voraussetzungen auszugehen hat als bei den dieser Arbeit zugrundegelegten heute gebräuchlichen Formen.

IV. *Softwarebestandteile mit üblicherweise nur mittelbar wahrnehmbarer Verkörperung auf magnetischen Datenträgern*

Die unter dieser Überschrift einzuordnenden Materialien sind ebenfalls in den Katalog aufzunehmen, weil der Umstand, dass deren Träger im gebräuchlichsten Zustand nur elektrische Impulse bzw. bei Lochkarten und dergleichen in solche Impulse umwandelbare Medien sind, nur als weitere Modalität der Festlegung zu betrachten ist. Für die urheberrechtliche Beurteilung gilt⁹³, dass die Aufnahme eines Werks auf magnetische Träger, dessen Einspeicherung oder Umwandlung in Impulse, jede mit Hilfe des Computers veranstaltete Umgestaltung in eine den menschlichen Sinnen unmittelbar zugängliche Form (Ausgabe auf Drucker oder Sichtgerät) wie auch sonst «jede körperliche Festlegung, gleichgültig, ob eine Wahrnehmung des Werks durch die menschlichen Sinne unmittelbar oder nur mittels Einschaltung eines Gerätes möglich ist⁹⁴», unter den Begriff der Vervielfälti-

91 Diagnosemeldungen sind die beim Programm-Ausdruck angebrachten umgangssprachlichen Fehlervermerke. Diese Vermerke werden vom Umsetzungsprogramm beifügt, wenn Fehler im Programm festgestellt werden.

92 CONTU, S. 38: «There have been great changes ... in the media in which programs are recorded. (...) It should be emphasized that these developments reflect differences only in the media in which programs are stored and not changes in the nature of the programs themselves.»

93 Eingehender zu diesem Problem unten Kap. 6 F I.

94 ULMER E., GRUR 1971, S. 300; in gleichem Sinne MÖHRING PH., GRUR 1967, S. 273f; GOTZEN F., DdA 1977, S. 18f mit Hinweisen; HUBMANN H., 4. Aufl., S. 132 und

gung fällt. Die besondere Art der physischen Existenz auf magnetischen Medien beeinträchtigt nicht die Schutzfähigkeit dieser Softwareergebnisse, weil sie stets den menschlichen Sinnen direkt zugänglich gemacht werden können und oft auch müssen. Dieser Gruppe sind einige bereits früher erwähnte Ergebnisse nochmals zuzuordnen, u.a. das Source- und Objektprogramm. Neu kommen hinzu:

1. *Referenz-, Parameter-, Textdateien und sonstige Dateien mit Steuerungsinformationen*

Charakteristisch für derartige Dateien ist ihre programmergänzende oder -ersetzende Funktion. Ihre Aufgabe besteht darin, bestimmte Programme genauer zu spezifizieren oder bestimmten Programmen einen Text- oder sonstigen Informationsvorrat zur Verfügung zu stellen, auf den sie bei gewissen Bedingungen zurückgreifen können. Diese Dateien und die ihnen zugehörigen Programme bilden damit stets eine gewisse Einheit. In den meisten Fällen kommt diesen Dateien kein von den Programmen unabhängiger Eigenwert zu. Je nach Funktion sind zwei Ausdrucksmittel vertreten: Bei den Textdateien handelt es sich um umfangreiche umgangssprachliche Elemente; bei den anderen Dateien sind ähnliche Charakteristiken vorhanden wie bei den Sourceprogrammen, in einfacheren Fällen allerdings beschränkt auf wenige Code- oder Grössenangaben.

2. *Testdaten*

Im Gegensatz zu den Verarbeitungsdaten, die zum Gegenstand und nicht zur Lösung einer Datenverarbeitungsaufgabe gehören, sind die Testdaten ein wichtiges Ergebnis der Softwareentwicklung. Mit Hilfe einer arbeitsaufwendigen Sammlung von Datenkonstellationen wird dafür gesorgt, dass die Soll-Funktionen der Programme rationell geprüft werden können. Es wird im Rahmen dieser Arbeit verzichtet, auf diese wenig gefährdeten Objekte einzutreten.

Zitate S. 131 und unten Kap. 6 F I. Es bieten sich ohne weiteres Analogien an zur Einspeicherung und Wiedergewinnung urheberrechtlich geschützter Werke auf Computer-Anlagen, ferner auch zur Fixierung von Musikwerken auf Musik- oder Piano-rollen, Schallplatten usw., wo ebenfalls nur mittels eines Gerätes das Werk sinnlich unmittelbar wahrnehmbar gemacht werden kann. Vgl. zur etwas abweichenden Anschauung in den USA unten Kap. 6 F I; ferner CONTU, S. 38: «Both recorded music and computer programs are sets of information in a form which, when passed over a magnetized head, cause minute currents to flow in such a way that physical work gets done.»

3. Kapitel: Beurteilung der rechtlichen Schutzerfordernisse aufgrund der Schutzinteressen der Beteiligten und der ausserrechtlichen Schutzmöglichkeiten

A. Entwicklungen in der Softwareherstellung und -vermarktung¹

Ein Schutz der Software ist nur dann erforderlich, wenn neben dem Hersteller auch Dritte daraus einen Nutzen ziehen können. Das ist der Fall, wenn die Software mehrfach verwendbar ist² und deshalb von den Anwendern der aufwendigen, den Bedürfnissen allerdings oft besser entsprechenden Eigenentwicklung oder Anpassung³ vorgezogen wird. Die Entwicklung der letzten Jahre zeigt deutlich, dass das Angebot und der Erwerb von mehrfach verwendbarer Software, vor allem im kommerziellen Bereich⁴, stark⁵ zunimmt⁶.

- 1 Vgl. JANSEN D., Diss. Hamburg 1972; NEUGEBAUER/DEHN/PHILIPP/WALTER, Forschungsbericht DV 76-04, München 1976 und dort auf S. 179ff zitierte Literatur; Diebold Management Report, München 1975.
- 2 Übliche Bezeichnung: Mehrfach verwendbare Software, Standard-Software, Programm-Pakete, Fremdsoftware oder Standardprogramme.
- 3 NEUGEBAUER/DEHN/PHILIPP/WALTER, S. 66ff. Danach wird «von allen Anwendergruppen die eigenerstellte Anwendungssoftware bezüglich der Problemanpassung und technischen Problemlösung mit deutlichem Abstand besser beurteilt als Fremdsoftware.» Dieses Ergebnis dürfte künftig (nach 1976) für die Fremdsoftware jedoch wesentlich günstiger ausfallen unter Berücksichtigung der geringen Angebotskenntnisse, fehlenden Vergleichsmassstäbe für Anwender sowie der Tatsache, dass die Qualität der Standardsoftware dank zunehmender Herstellungserfahrung, neuer Techniken und der Öffnung des Softwaremarktes (vgl. betreffend Unbundling unten Ziff. 3) stets deutlich verbessert werden konnten.
- 4 Institut, Bestand Schweiz 1978, Tab. 46, S. 51: 60 bis 80% der Computeranwender setzen den Computer für die wichtigsten kommerziellen Aufgaben ein: 78,4% Rechnungswesen, 71,1% Fakturierung, 68,4% Lohn- und Gehaltsabrechnung, 68,2% Statistik, 63,2% Lagerkontrolle usw. Die deutliche Konzentration auf einzelne Aufgabengebiete ist ein Umstand, der für eine weitere Verbreitung der Standardsoftware spricht.
- 5 Nach NEUGEBAUER/DEHN/PHILIPP/WALTER, S. 159 und 161, verdoppelt sich in der BRD zwischen 1975 und 1980 das Nachfragepotential für mehrfach verwendbare Anwendungssoftware in der gewerblichen Wirtschaft. Der 1975 auf mehrfach verwendbare Software entfallende Umsatzanteil an der gesamten extern erstellten Anwendungssoftware von 45% wird sich bis 1980 auf rund 60% erhöhen. Dieser Trend wird noch dadurch begünstigt, dass als Schwerpunkt der geplanten Automatisierungsprojekte bei allen Anwendergruppen der Bereich der kommerziellen Software, insbesondere Aufgaben des Rechnungswesens, festgestellt wurde. Wesentlich ausgeprägtere gleichlaufende Entwicklungen wurden in den USA beobachtet (vgl. CONTU, S. 38f). Zusätzliches Gewicht erhält dieser Trend, weil wegen der starken Verbilligung der Hardware in den letzten Jahren die Zahl der Computer «dramatisch» zunimmt und damit auch die Anzahl der für sie erforderlichen Programme.
- 6 Aufgrund von eigenen mehrjährigen Beobachtungen bei einem Hardware-Hersteller kann eine derartige Bevorzugung bei allen Unternehmensgrössen festgestellt werden.

Die wichtigsten Gründe für diese Entwicklung zugunsten der mehrfach verwendbaren Software und deren Verselbständigung zu einem verkehrsfähigen wirtschaftlichen Objekt⁷ dürften sein:

1. Die Mehrfachverwendung ermöglicht die Verteilung von hohen Entwicklungskosten⁸ auf verschiedene Anwender und reduziert die Wahrscheinlichkeit von überflüssigen Parallelentwicklungen und Wartungsaufwendungen. Für den einzelnen Anwender erlaubt die raschere maschinentechnische Implementierbarkeit der Standardsoftware und die Verkürzung des Test-, Fehlerbehebungs- und Wartungsverfahrens gegenüber «hausgemachten» Programmen, rascher eine hohe Auslastung und produktivitätssteigernde Nutzung der Hardware zu erreichen. Das Risiko des Computereinsatzes wird in der Regel reduziert, weil auf seiten des Anwenders neben den betrieblich-organisatorischen Anpassungsproblemen nicht noch gleichzeitig die gesamten EDV-technischen Fragen bewältigt werden müssen⁹.

2. Wie oben gezeigt¹⁰, ist eine starke Konzentration auf gleichartige Anwendungsgebiete zu beobachten. Die auf diesen Gebieten vorhandene grosse Verwandtschaft von Problemstellungen¹¹ ist ein wichtiges Indiz für das grosse Marktpotential der Standard-Software. Der Gebrauch der Standard-Software wird zudem stark begünstigt durch die Verbilligung der Hardwarekapazitäten: Die Mehrfachverwendbarkeit wird erreicht, indem bei der Softwareentwicklung eine Breite von Benutzerbedürfnissen berücksichtigt wird. Das ist aus verschiedenen Gründen meist mit Redundanzen und Effizienzverlusten verbunden. Je mehr sich aber der Anteil der Aufwendungen für Software gegenüber demjenigen für Hardware vergrössert, desto eher können diese Nachteile in Kauf genommen werden¹².

7 Noch Ende der sechziger Jahre betrachteten z.B. KÖHLER R., S. 8, und SIDLER P., S. 5ff, die Software vor allem als «Bestandteil» elektronischer Datenverarbeitungsanlagen.

8 Institut, Bestand Schweiz 1978, Tab. 48, S. 53, zeigt, dass von rund 2200 Anwendern (Tab. 38, S. 43) je Computeranwender pro Jahr durchschnittlich Fr. 218 000 für die Softwareentwicklung und Fr. 73 000 für die Programmwartung und -pflege aufgewendet wurden.

9 Zufolge einer 1972 von der Düsseldorfer Unternehmensberatungsfirma A.T. Kearny und der Zeitschrift manager-magazin (manager-magazin 1972, Heft 9, S. 58ff, zitiert in SCHWAIGER/KOCKLER S. 353) erstellten Studien über Kosten, Nutzung und Management von EDV-Anlagen werden in der BRD «Milliarden verschwendet», weil die erwähnten Vorteile der Standardsoftware nicht genügend ausgeschöpft werden. Auch wenn diese Angaben etwas hoch erscheinen, darf angenommen werden, dass der Standard-Software in grossem Masse die Aufgabe zukommt, unnütze Aufwendungen zu vermeiden.

10 Oben Kap. 3 A.

11 Besonders bei kommerziellen Applikationen gelten für viele Betriebe ähnliche betriebswirtschaftliche Grundlagen (Buchhaltungs-, Kostenrechnungs-, Entlohnungssysteme usw.), gleiche gesetzliche Auflagen (Vorschriften für die Rechnungslegung, Steuern, Sozialabgaben usw.) und revisions- oder arbeitstechnische Anliegen.

12 Bei der Softwareherstellung konnten in letzter Zeit bei weitem nicht die gleich grossen Produktivitätsfortschritte erzielt werden wie bei der Hardwareherstellung. Nach den von SCHWAIGER/KOCKLER, S. 354, zitierten Untersuchungen veränderte sich das

3. Für die Entwicklung eines Softwaremarktes war es vorteilhaft, dass seit Ende der sechziger Jahre die Hardwarehersteller nach und nach den Erwerb aller Programm-Typen aus der Koppelung mit dem Hardware-Vertrag herauslösten (sog. Unbundling¹³) und in der Folge für die Software separate Lizenzverträge abgeschlossen werden konnten. Dies erleichterte den Zugang zum Markt für reine Softwarehersteller¹⁴. Die Anwender können dank der eingetretenen Verselbständigung des Software-Marktes freier die bestgeeignete Software auswählen¹⁵ und besser, d.h. ohne die beim Bundling vorhandene schlecht vergleichbare Vermischung der Leistungsarten, die Angebote mit anderen Hardware- und Softwareherstellern vergleichen. Angesichts der heutigen Dimensionen des Software-Marktes wird oft von einer «Informationsindustrie» gesprochen.

4. NEUGEBAUER/DEHN/PHILIPP/WALTER halten ferner als Umfrageergebnis fest: «Ein beträchtlicher Teil der Anwender kann die geplanten Automatisierungsprojekte in der vorgesehenen Zeit nicht ausschliesslich mit vorhandenem Personal durchführen. Bei einem beabsichtigten Realisierungszeitraum von durchschnittlich zwei Jahren würden die Anwender ca. vier Jahre beim vorhandenen Personalbestand benötigen.»¹⁶ Dieses Bild ändert sich kaum, wenn die bekannten Rationalisierungsmethoden¹⁷ zur Erhöhung der Kapazität vorhandener DV-Kräfte voll ausgeschöpft würden, weil «die wachsende Bindung von Personalkapazität durch Wartungs- und Umstellungsarbeiten diese «Lücke» noch vergrössern wird.»¹⁸ Die Standard-Software scheint für die Deckung des wachsenden Softwarebedarfs¹⁹ die

Verhältnis an den Gesamtaufwendungen für EDV von ursprünglich 40% Software; auf 70% und 60% Hardwarekosten auf 30%, wobei im Einzelfall erhebliche Abweichungen möglich sind. Diese für 1973 festgestellten Verhältniszahlen berücksichtigen noch nicht die in der Zwischenzeit eingetretenen weiteren massiven Reduktionen bei den Hardwarepreisen.

13 Siehe hierzu FUCHS A., S. 80ff; LEHMANN B., sysdata 11/78, Abschn. VII.

14 Den Softwaremarkt teilt heute eine ältere Anbietergruppe, bestehend aus Hardwareherstellern und Anwendern, die durch den Verkauf der primär für eigene Zwecke entwickelten Software einen Kostenbeitrag hereinholen wollen, und eine jüngere Gruppe von reinen Softwareherstellern, die gezielt nach Absatzchancen Software herstellt und samt den erforderlichen Zusatzleistungen wie Installation, Wartung, Beratung, Schulung und dergleichen vertreibt. Im Jahre 1976 waren in der BRD bereits etwa 250 wichtigere Anbieter von Standard-Software im kommerziellen und technisch-wissenschaftlichen Bereich bekannt. Nicht berücksichtigt sind Verbände und «anbietende Anwender», vgl. NEUGEBAUER/DEHN/PHILIPP/WALTER, S. 140ff und insbes. S. 143.

15 Hersteller, Software-Häuser oder besondere Institute übernahmen die Funktion einer «Clearing-Stelle für Software». Einen Überblick über das grosse Angebot vermittelt z.B. der ISIS Software Report, z.Zt. 10. Jahrgang, München 1979, oder ZELLE/SCHNECHTER/SCHMID, Programmbibliothek für Wissenschaft und Verwaltung, Wien 1975.

16 NEUGEBAUER/DEHN/PHILIPP/WALTER, S. 63.

17 Siehe näheres ebd., S. 61.

18 Ebd.

19 Ein grosser Softwarebedarf dürfte allein wegen des starken Wachstums des Computerbestandes entstehen (vgl. Institut, Bestand Schweiz 1978, Tab. 81, S. 93: Voraussage Computerbestand).

einzigste Alternative zu sein, weil der vorhandene Personalbestand sich – zumindest kurz- bis mittelfristig – kaum erhöhen lässt, denn qualifizierte DV-Kräfte sind «weder auf dem Arbeitsmarkt leicht zu bekommen noch gegenwärtig ausreichend in den Anwenderunternehmen vorhanden²⁰.» Diese Feststellung lässt sich ohne weiteres auf die Schweiz übertragen.

Zusammenfassung

Die Bedeutung der Computersoftware als verselbständigtes und verkehrsfähiges wirtschaftliches Objekt wird weiterhin stark zunehmen²¹. Der wachsende Softwarebedarf wird in zunehmendem Masse²² von Anbietern gedeckt, die sich ausschliesslich mit der Herstellung von Software befassen²³. Der erstrebte Markterfolg ihrer Leistung kann gefährdet werden,

20 NEUGEBAUER/DEHN/PHILIPP/WALTER, S. 64.

21 Marktprognosen, die sich auf Beobachtungen in den USA stützen, rechnen gar, dass im Laufe der achtziger Jahre über 50% der Gesamtausgaben der EDV für Standard-Software aufgewendet werden. Vgl. NEUGEBAUER/DEHN/PHILIPP/WALTER, S. 149f.

22 Bereits klar erkennbar ist auch in Europa der von der CONTU, S. 40, für die USA festgestellte Trend: «If present computer industry trends continue, it is all but certain that programs written by non-machine manufacturers will gain an increasing share of the market, not only because writing programs and building machines are two very different skills that need not necessarily concur, but also because program writing requires little capital investment.»

23 Bei der Begründung der Empfehlungen für einen Softwareschutz im Rahmen der Revision des US-Copyright-Acts von 1976 sieht die Kommission (vgl. CONTU, S. 38f) zu Recht zahlreiche Parallelen mit den Entwicklungen und Problemen der Tonträgerindustrie: Zunächst waren hier die Schutzbedürfnisse ebenfalls gering, weil die Apparathersteller eigens für ihre Automaten geeignete Tonträger herstellten. Der Werkgeuss und die Vielfältigkeit war nur mit Hilfe dieser Apparate und den dafür vorgesehenen Tonträgern möglich. Die Kosten der Musikproduktion vereinigten sich untrennbar mit denjenigen für die Herstellung der Apparate und der physischen Tonträger. Erst als sich mit der Zeit eine selbständige Tonträgerindustrie entwickelte, deren Produktionen unabhängig von bestimmten Geräten oder Tonträgern verwertet und im Verhältnis zur ersten Festlegung mit geringem Aufwand dupliziert werden konnten, offenbarte sich die Verletzbarkeit ihrer Leistungen. Der hohe Aufwand für die Leistungserzeugung einerseits und die leichte Verletzbarkeit dieser Leistungen andererseits liess eine Interessendichte erkennen, die der Gesetzgeber – vorläufig nur im Ausland – nicht übersehen durfte.

Etwas zeitlich zurückversetzt zeichnet sich die gleiche Entwicklung in der Computerindustrie ab: Aus der Hardwareindustrie (entsprechend Apparateindustrie) heraus verselbständigt sich die Softwareindustrie, deren Erzeugnisse mit zunehmender Kompatibilität losgelöst von bestimmten Computern oder Datenträgern verwertet und verbreitet werden können. Das Erzeugnis verselbständigt sich mehr und mehr und löst sich von einer bestimmten Nutzungseinrichtung und deren Träger. Der immaterielle Charakter und die potentielle Ubiquität treten vermehrt in den Vordergrund. Genauso wie sich für die Tonträgerindustrie schliesslich die Probleme in aller Deutlichkeit zeigten, als billige Tonbandgeräte und Tonträger massenweise auf den Markt kamen, ist im Ansatz bereits die gleiche Situation für die Softwareindustrie erkennbar mit dem Vormarsch von billigen, unproblematischen Computern, die sich durch grössere Kompatibilität auszeichnen und die ebenfalls entsprechende Kopiereinrichtungen haben. Wenn wir vom Bereich der bei Home-Computern und in der Unterhaltungselektronik (electronic toys) eingesetzten Software (Fernseh-, Kleincomputerspiele usw.) absehen, wo sich die Situation kaum mehr unterscheiden dürfte, werden die Softwarehersteller im allgemeinen vorläufig besser geschützt bleiben, weil die Komplexität ihres Leistungsgegenstandes dessen Übernahme stark hemmt.

weil ohne Wiederholung des Aufwandes das von einem grossen Anwenderkreis nachgefragte Leistungsergebnis mit technischen Mitteln auf billige und einfache Weise von Dritten kopiert werden kann. Die Interessen der Leistungserbringer werden dadurch in zweifacher Hinsicht berührt: Das Duplikat schmälert die Absatzchancen der kopierten Software, zudem kann es in Konkurrenz zur ursprünglichen Software auf den – wie gezeigt bereits vorhandenen und stark expandierenden – Markt gelangen.

Die aufgeführten Sachverhalte decken Interessen auf, die grundsätzlich als schutzwürdig bewertet werden müssen.

B. Der «Selbstschutz» und Massnahmen zum Schutz der Software

Dem «Selbstschutz» und den Massnahmen zum Schutz der Software (sog. technologischer Schutz) werden in der Literatur neben den rechtlichen Schutzmöglichkeiten zu Recht eine eigenständige und wesentliche Bedeutung zugemessen²⁴.

Zum «Selbstschutz» gehören Verbreitungshindernisse, die ohne besonderes Zutun aus der Eigenart der Software hervorgehen. Bei den Schutzmassnahmen handelt es sich um Methoden, die Software analog dem Schutzobjekt «Daten»²⁵ vor Verletzungen bewahren können.

Diese Schutzmassnahmen können dem Gebiet der Datensicherung zugeordnet werden²⁶.

Falls der «Selbstschutz» und die Datensicherung das tatsächliche Verletzungsrisiko mit zumutbarem Aufwand gering halten können, so setzt der Vertrieb von Software «nicht denknotwendigerweise einen urheber- oder wettbewerbsrechtlichen Rechtsschutz voraus²⁷.» Für die Beurteilung des Verletzungsrisikos sind nur die allgemein zugänglichen Schutzmassnahmen in Betracht zu ziehen. Nicht zu berücksichtigen sind Schutzvorkehrungen, die nur vom Hardware-Hersteller eingerichtet werden können²⁸. Fol-

24 Vgl. KOLLE B., GRUR 1973, S. 617 und dort zitierte Literatur.

25 Wie oben in Kap. 2 A erwähnt, unterscheiden sich Verarbeitungsdaten und Softwarebestandteile in der Speicherungsform nicht.

26 DIERSTEIN/FIEDLER/SCHULZ, S. 127f: Mit Datensicherung – oft auch mit dem irrtümlicherweise synonym behandelten allgemeineren Begriff des Datenschutzes – werden die Gesamtheit der Massnahmen bezeichnet, mit denen gespeicherte Daten (dazu gehören auch gespeicherte Softwarebestandteile) vor Verlust, unbefugter Veränderung oder Benutzung gesichert werden.

27 SCHWAIGER/KOCKLER, S. 368.

28 Der Hardware-Hersteller kann z.B. statische Softwarekomponenten auf Festwertspeichern oder anderen Datenträgern fixieren, damit diese nur auf einer bestimmten Zentraleinheit benutzt oder nur mit sehr aufwendigen technischen Einrichtungen dupliziert werden können.

Wie bereits oben festgestellt (Kap. 3 A), vergrössert sich die Zahl der reinen Software-Hersteller. Sie können nicht zu hardwaremässigen Schutzmassnahmen greifen. Weil aber gerade bei ihnen das Schutzinteresse besonders ausgewiesen ist, dürfen die ihnen nicht offenstehenden Schutztechniken nicht in die Beurteilung einbezogen werden.

gende Verbreitungshindernisse stehen sowohl den Hardware- wie auch den reinen Softwareherstellern zur Verfügung: Die komplexe Natur des Schutzgegenstandes und der sich dadurch ergebende «Selbstschutz», administrativ-organisatorische sowie softwaremässige Datensicherungsmassnahmen und Vorsichtsmassnahmen bei der Verbreitung der Software²⁹.

I. Der «Selbstschutz» der Software

Programme, vor allem mehrfach verwendbare Programmpakete sind sehr komplexe, für einen einzelnen kaum in allen Einzelheiten überblickbare Gebilde. Daraus ergeben sich Verbreitungshindernisse, die unter zwei Gesichtspunkten zu betrachten sind.

1. Eignungsprüfung und Realisierung des praktischen Einsatzes

Nur in wenigen Fällen genügt die Übernahme der leicht kopierbaren wertvollsten Softwareteile³⁰, um die Software auf ihre Eignung für eine bestimmte Problemstellung zu prüfen oder gar durch Fachleute, die mit den Programmen unvertraut sind, für den praktischen Einsatz einzurichten. Die Programme enthalten die grösste Detaillierung der Problemlösung und sind deshalb nur nach umfangreichem und kostspieligem Analyseaufwand inhaltlich zu erfassen³¹. Der zu erwartende Aufwand ist im allgemeinen grösser, als der ordentliche Programmierwerb samt zugehöriger Dokumentation, Instruktion und Installationsunterstützung kostet.

Die Kopie des Programm-Codes bringt also kaum einen Nutzen, wenn nicht gleichzeitig exakte Systeminformationen und Gebrauchsanweisungen greifbar sind. Das können separate Beschreibungen, Dokumentationen, mündliche Erläuterungen oder im Programm enthaltene Anleitungen sein³². Letztere sind entweder als umgangssprachlicher Kommentar innerhalb des Sourceprogrammes oder – meistens in Dialogprogrammen³³ – als selbsterklärende Programmfunktionen auch im Objektprogramm enthalten. Der wesentliche Aspekt des hier gemeinten «Selbstschutzes» besteht demnach darin, dass Programme ohne die Begleitung von umfangreichen – im allgemeinen in eine urheberrechtlich schützbarer Form gebrachte – Erläute-

29 Im einzelnen nachfolgend Abs. I bis III.

30 Vor allem der Code der Source- und Objektprogramme.

31 Vgl. KOLLE G., GRUR 1973, S. 616 mit Hinweisen.

32 Es handelt sich in der Regel um umgangssprachliche Schriftwerke, denen in der Regel der urheberrechtliche Schutz zusteht. Vgl. oben Kap. 2 E I und unten Kap. 6 A.

33 Dialog: Austausch von Frage und Antwort zwischen Datenstation (meist Bildschirm) bzw. Anwender und EDV-Anlage. Der Anwender wird Schritt um Schritt vom Programm «geführt», sog. leadthrough.

rungen praktisch wertlos sind. Dieser Schutz müsste dann weitgehend entfallen, wenn diese Erläuterungen gleichzeitig mit dem Programmcode zugänglich sind, was oft für die Gattung der immer gebräuchlicher werdenden Dialogprogramme zutrifft. Der Anwender eines Dialogprogrammes ist bei diesen in der Regel nur noch wenig auf separate Begleitinformationen angewiesen, denn diese Programme können im Gegensatz zu den früher üblichen Stapelverarbeitungsprogrammen weitgehend selbsterklärend ausgestaltet werden.

Sie geben selber auf dem Bildschirm in der Umgangssprache Auskunft über Art, Inhalt und Bedingungen von Eingabedaten und Verarbeitungsergebnissen.

Der hier besprochene, oft hoch eingeschätzte Aspekt des «Selbstschutzes» darf bei unkomplizierten oder selbsterklärenden Dialogprogrammen nicht überbewertet werden.

2. *«Selbstschutz» wegen Fehlerhaftigkeit und Notwendigkeit dauernder Programmwartung*

Programme mit Tausenden von Befehlen können kaum fehlerfrei sein. Regelmässig werden während der mehrjährigen Einsatzdauer³⁴ Korrekturen, Verbesserungen oder Anpassungen an neue Gegebenheiten der Problemstellung³⁵, Hardware³⁶ oder Systemsoftware³⁷ erforderlich. Die dauernde Wartung ist unerlässlich für die kontinuierliche und optimale Nutzung von hochentwickelten Programm-Paketen³⁸. Dem Benutzer bringt aus folgenden Gründen die unautorisierte Verwendung von Programmen auf die Dauer keinen Nutzen:

– Einer der wichtigsten Vorteile des Einsatzes von Standard-Software besteht darin, dass der Programmierer vom Hersteller die ständi-

34 Unten Kap. 6 F III und 7 B VII.

35 Bei einem Lohnabrechnungsprogramm z.B. die Änderung des Abrechnungsmodus für Sozialabgaben.

36 Anpassungen bei Konfigurationsänderungen und Wechsel des Maschinenmodells.

37 Freigabe von verbesserten oder ergänzten Betriebssystemprogrammen (Software-Releases), was unter Umständen eine Anpassung der hierarchisch untergeordneten Anwendungsprogramme bedingt (vgl. oben Kap. 2 B I 2).

38 Derartige Überlegungen dürften auch dem Vorschlag der CONTU, S. 45, zugrunde liegen, die als Ergänzung der Section 117 des US-Copyright-Acts von 1976 folgende gesetzliche Einschränkungen des Exklusivrechtes an Programmen vorgesehen hatte: Recht des autorisierten Inhabers eines Programmduplikates zur Herstellung von weiteren Duplikaten (für Sicherungs- und Archivzwecke) und zur Bearbeitung des Programmes (Anpassungen, Verbesserungen und Übersetzungen in andere Programmiersprachen), vorausgesetzt, «that such a new copy or adaptation is created as an essential step in the utilization of the computer program ...». Diese Rechte würden im wesentlichen nur im Rahmen des Eigengebrauches zugestanden. Es handelt sich um Befugnisse, die in unserem System durch den Vorbehalt des privaten Werkgebrauches bereits eingeräumt wären.

gen Dienstleistungen für Fehlerbehebung und Wartung in Anspruch nehmen kann und er nicht – falls er überhaupt im Besitze der Source-Programme und Dokumentationen ist – im kostspieligen Alleingang in den unvertrauten Programmen die anspruchsvollen und aufwendigen Änderungsarbeiten vornehmen muss. Es versteht sich, dass nur der rechtmässige Programmiererwerber diese Leistungen beanspruchen kann.

– In den meisten Fällen muss ein mehrjähriger ununterbrochener Einsatz eines Programmpaketes gewährleistet sein, damit die im Vergleich zu den Programmiererwerbskosten oft viel grösseren Aufwendungen für die Programminstallation und die betrieblich-organisatorischen Anpassungen in tragbarem Rahmen abgeschrieben werden können. Für die Benutzer, die auf einen kontinuierlichen Einsatz des Programmes angewiesen sind, bringt die unberechtigte Programmbenützung oder -weitergabe trotz der unsicheren Rechtslage zu grosse Risiken mit sich, wenn sie jederzeit mit einer vorzeitigen Beendigung des verletzten Programmlizenz- und Wartungsvertrages oder mit einer Klage auf Unterlassung des weiteren Programmgebrauches rechnen müssen.

– Die Wartung von Programmen kann kaum auf rationelle Art und Weise von Anwendern durchgeführt werden, die nur über die üblicherweise beim Programmiererwerb abgegebenen Anwender-Dokumentationen verfügen und nicht Zugang zu den beim Software-Hersteller verbleibenden detaillierten Systemunterlagen haben. Voraussetzung wäre überdies, dass der Anwender überhaupt über die notwendigen Source-Programme verfügen könnte³⁹.

Der «Selbstschutz» ist unter diesen Gesichtspunkten sehr gross und verringert das Verletzungsrisiko erheblich. Das gilt nicht für Softwareprodukte aus dem Gebiet der Unterhaltungselektronik (electronic toys) und für Programme, die nur für einen gelegentlichen oder kurzfristigen Einsatz vorgesehen sind.

II. Organisatorisch-administrative und softwaremässige Massnahmen zum Schutz der Software (Datensicherung)

Diese Massnahmen bezwecken, die auf einer EDV-Anlage oder in Archiven gespeicherten Softwarebestandteile von einer unautorisierten Nutzung, Vervielfältigung, Weitergabe oder Veränderung zu schützen. Alle diese Massnahmen folgen dem Prinzip, dass bei jedem «kritischen» Verarbeitungs- oder Kopiervorgang administrative Kontrollverfahren einsetzen oder auf dem Computer zwingend zu durchlaufende Softwarekomponenten aktiviert werden, die aufgrund eines Erkennungsverfahrens zwischen autori-

sierten und nichtautorisierten Benutzerhandlungen zu unterscheiden haben und bei erkennbarer Gefährdung der gespeicherten Softwarebestände diese unverzüglich sperren oder unbrauchbar machen, falls sie nicht bereits vor Beginn des Erkennungsverfahrens in verschlüsselter oder sonstwie gebrauchsunfähigem Zustand sind⁴⁰.

Aus der Vielzahl der bekannten Methoden sind etwa zu erwähnen:

1. Durchführung eines Erkennungsverfahrens mit Passwörtern (Durchgangswörtern).

2. Einrichtung von Zähl- und Durchlaufkontrollen, die nur eine erlaubte Anzahl von kritischen Vorgängen zulassen, diese dokumentieren und, falls die Zahl überschritten wird, die Daten unbrauchbar machen.

3. Feste Koppelung von Hardwarebestandteilen und Software mit gegenseitig verknüpften Identitätsmerkmalen, so dass die Software nur in Verbindung mit bestimmten Computern oder Datenträgern verwendet werden kann.

4. Hardware- und/oder softwaremässige Kontrollverfahren zur Überwachung der Nutzungsfrequenz und Nutzungsart.

5. Administrativ-organisatorische Massnahmen, wie Auswahl von vertrauenswürdigen Personal, Zugriffs- oder Zugangssperren bei Softwarebibliotheken und Operating-Konsolen usw.

Diese Methoden eignen sich nicht überall und jederzeit: Beschränkter sind die Möglichkeiten allgemein bei kleineren Computersystemen, wo derartige Kontrollsysteme aus Kapazitätsgründen oder wegen Beschränkungen der Betriebssoftware nicht in wünschbarem Umfang realisierbar sind. Kaum zu verwirklichen oder aber stark behindernd sind gut eingespielte Kontrollen in der Entwicklungs- und Testphase. Schwieriger und sehr aufwendig sind derartige Kontrollen auch, wenn die Programmpakete stark verbreitet sind oder auf Computern verschiedener Hardware-Hersteller angewendet werden.

Unbestritten ist, dass die Kosten der Sicherungsmassnahmen und die in Kauf zu nehmenden Behinderungen beträchtlich sind, «die Quantifizierung des Nutzens von Schutzmassnahmen sehr schwierig oder gar unmöglich ist⁴¹» und es «auch bei Einsatz grösster finanzieller Mittel keinen absoluten, alles umfassenden Schutz, weder gegen Verlust, Zerstörung und Beschädigung von Daten (einschliesslich der Software) noch vor Missbrauch der Datenverarbeitung, gibt⁴².» Obwohl der mit grossem Aufwand erreichbare Schutz beträchtlich ist, darf nicht gefolgert werden, dass ein

40 Vgl. ausführlicher DIERSTEIN/FIEDLER/SCHULZ, insbes. S. 126–268 und dort zitierte Literatur. Aus der Sicht eines Computerherstellers: LÜTHY H. R., NZZ Nr. 568 vom 6. Dezember 1973.

41 DIERSTEIN/FIEDLER/SCHULZ, S. 227.

42 Ebd.

wirksamer und lückenloser Schutz rein praktisch, ohne rechtliche Rücken-
deckung gelöst werden kann. Zu Bedenken ist immerhin, dass die Software-
Hersteller mit oder ohne wirksamen Rechtsschutz kaum von den aufwendi-
gen Sicherungsmassnahmen entlastet werden könnten, denn «selbst wenn
für den Bereich der Software ein ausreichender Rechtsschutz entwickelt sein
wird, werden nur genügende Kontrollsysteme die Rechtsverletzungen fest-
stellbar machen⁴³.»

III. *Vorsichtsmassnahmen bei der Verbreitung der Software*

Zur Vermeidung unautorisierter Nutzungen oder Weitergaben wur-
den in der Praxis «verschiedene Massnahmen als Handelspraktiken entwik-
kelt⁴⁴»:

– «Es werden nur sehr schmale oder überhaupt keine Programmdo-
kumentationen überreicht⁴⁵.» Wie bereits erwähnt⁴⁶, kann nur bei gleichzei-
tiger Verfügbarkeit von Programm-Code und umfangreicheren Dokumen-
tationen die Software auf ihre Eignung geprüft und selbständig für den
Gebrauch eingerichtet werden. Aufgrund der üblicherweise abgegebenen,
wenig aussagekräftigen Bedienungsanleitungen kann in der Regel nur ein
für den konkreten praktischen Einsatz bereits vorbereitetes Programmpaket
angewendet werden. Einführungs- und Anpassungsarbeiten bleiben dem
Software-Hersteller vorbehalten⁴⁷.

– Meistens werden nur die Objekt- und die unbedeutenderen
Source-, nicht aber die für einen kontinuierlichen Einsatz und die Wartung
erforderlichen wichtigsten Sourceprogramme abgegeben. Fehlerbehebun-
gen, Anpassungen und Verbesserungen können somit im Normalfall nicht
vom Lizenznehmer ausgeführt werden.

– Softwareangebote werden meist beschränkt auf eine sehr allge-
mein gehaltene Beschreibung der Aufgaben und Ergebnisse der einzusetzen-
den Programme. Weitergehende Informationen des Anbieters oder detail-
lierte Lösungsvorschläge sind vor dem Programmwerb bereits kosten-
pflichtig und/oder werden nur bei vorheriger vertraglicher Zusicherung der
Diskretion abgegeben. Die Prüfung der zu erwerbenden Programme erfolgt
in der Regel unter Anleitung des Software-Herstellers. Mit Ausnahme der –

43 KOKAI T. R., S. 13.

44 KOKAI T. R., S. 12ff eingehend.

45 Ebd., S. 13.

46 Vgl. oben Abs. I.

47 Nach NEUGEBAUER/DEHN/PHILIPP/WALTER, S. 118, werden unter 5 % der insgesamt
angebotenen Software ohne Zusatzleistungen, wie Installation (Einrichtung der Pro-
gramme), Wartung, Schulung, Beratung usw., weitergegeben.

ohnehin urheberrechtlich nicht schützbaeren – konzeptionellen Ideen sind die Softwarebestandteile in der Angebotsphase kaum übernahmegefährdet⁴⁸.

Es ist demnach festzustellen, dass sich bei der Software-Lizenzierung Praktiken herausgebildet haben, die vorläufig – aber gelegentlich in behindernder Weise – den Schutzinteressen der Software-Hersteller weitgehend entgegenkommen. Das Funktionieren des Softwaremarktes dürfte insgesamt kaum beeinträchtigt werden. Nicht unproblematisch ist die durch derartige Praktiken geschaffene dauernde Abhängigkeit des Softwareerwerbers von den Dienstleistungen (Wartung, Anpassungsarbeiten, Programm-einrichtung usw.) des Softwarelieferanten. Es könnte eher erwartet werden, dass die Softwarelieferanten zum Vorteil der Erwerber diese Praktiken zurückhaltender anwenden würden oder müssten, wenn ihre Produkte durch ein wirksames rechtliches System geschützt wären.

Ergebnis

Die Gesamtheit der denkbaren Massnahmen und Umstände zum tatsächlichen Schutz der Software lässt das Verletzungsrisiko und das rechtliche Schutzinteresse als gering einschätzen⁴⁹. Anders ist es bei unkomplizierteren, einfach zu handhabenden oder den zahlreicher werdenden, selbst-erklärenden dialogorientierten Programmpaketen, wo der Anwender in kleinerem Masse auf Dienstleistungen des Softwarelieferanten oder umfangreiche programmbegleitende Dokumentationen angewiesen ist. Unter Berücksichtigung der Nachteile, die infolge der erläuterten Schutzmassnahmen von Softwarelieferanten und -erwerbern in Kauf zu nehmen sind, dürfte auf beiden Seiten das Interesse an einem wirksamen rechtlichen Schutz vorhanden sein.

C. Die Beurteilung der mit dem Softwareschutz verknüpften Interessen

Mehrere Autoren haben bereits ausführlich die mit dem Softwareschutz verwickelten Interessen der Hardwarehersteller, der reinen Software-

48 Weiteres bei LEHMANN B., sysdata 8–9/78, S. 2.

49 Im wesentlichen übereinstimmend die Ergebnisse des Colloque de Strassbourg, insbesondere Beiträge von CUNNINGHAM W. und MOUSSERON J.-M., La protection des résultats de la recherche face à l'évolution des sciences et techniques, Paris 1969, S. 132ff., ferner KOLLE G., GRUR 1973, S. 616 und dortige Hinweise, etwas a.M. SCHWAIGER/KOCKLER, S. 396.

Es ist allerdings anzunehmen, dass diese Autoren die in jüngerer Zeit sichtbar gewordenen Entwicklungen auf den Gebieten der benutzerfreundlichen, in der Handhabung problemloseren dialogorientierten Computer und der Unterhaltungselektronik noch nicht berücksichtigen konnten.

hersteller, der Softwareerwerber und der Allgemeinheit definiert und bewertet, was für die realistische Würdigung dieses Sachverhaltes nur nützlich ist⁵⁰. Diesen Ausführungen gibt es mit Ausnahme der folgenden Gesichtspunkte nichts mehr hinzuzufügen:

I. Zum Interessenverhältnis von Hardware- und reinen Softwareherstellern (Softwarehäusern)

Der Grundtenor lautet hier: «... wie in allen Ländern mit hoher technologischer Entwicklung stehen die Interessen der Hardwarehersteller und Softwarehäuser entgegengesetzt gegenüber. Die Hardwarehersteller haben kein erhebliches Interesse an einem Rechtsschutz für Computersoftware. Die Software ist für die Computerhersteller nur ein Verkaufsargument. Das Interesse der Softwarehäuser an einem wirksamen Rechtsschutz ist hingegen sehr weit entwickelt, da auf dem deutschen Markt die Möglichkeiten, Software für viele Computertypen und Konfigurationen zu nutzen, von erheblicher Bedeutung sind⁵¹.» Ferner wird betont, die Softwarehäuser seien in besonderem Masse auf ein gesichertes Entgelt angewiesen, weil die Veräußerung der Software die alleinige Grundlage ihrer Existenz sei.

Wenn auch die Interessen der Computerhersteller an einem Rechtsschutz nach wie vor geteilt sind, so kann der oben zitierten Interessenbewertung in dieser Form nicht zugestimmt werden.

Eher uninteressant ist für die Hardwarehersteller in der Tat ein rechtlicher Schutz im Bereiche der Betriebssoftware, weil diese wegen ihrer starken hardwarespezifischen Ausrichtung in der Regel nur auf einem bestimmten Computertyp eines Herstellers⁵² eingesetzt werden kann und meistens gleichzeitig mit der Hardware abgegeben wird. Die Betriebssoftware wird auch ohne besonderen Schutz die gesicherte Domäne des Hardwareherstellers bleiben, weil die komplexe technische und wirtschaftliche Einheit von Hardware und Betriebssoftware kaum ein lohnendes Betätigungsfeld für reine Softwarehersteller sein kann. Der Hardwarehersteller hat in diesem Bereich kaum mit «Ausfällen» zu rechnen und der Absatz der Betriebssoft-

50 Näheres bei KÖHLER R., S. 31ff; ISELIN F., S. 30ff; BRAUN B., S. 1343; KOKAI T. R., S. 14ff; KOLLE G., GRUR 1973, S. 619ff und dortige Hinweise. Aus amerikanischer Sicht etwa CARNAHAN W. H., Protection of Computer Programs – a Dilemma, S. 42ff.

51 ISELIN F., S. 35; Ähnliche Auffassung von KOLLE G., GRUR 1977, S. 74, im Zusammenhang mit dem sehr weitreichenden, inzwischen weitgehend abgelehnten Patentschutz: «Der ... verweigerte Ausschliesslichkeitsschutz durch Patente verschafft insbesondere den Herstellern von Datenverarbeitungsanlagen den aus ihrer Sicht notwendigen Freiraum, da die dem Hardware-Absatz sicher förderliche umfassende Nutzungsmöglichkeit der Computer durch Drittrechte nicht eingeschränkt werden kann, und wird von ihnen sicher begrüßt werden.»

52 Siehe oben Kap. 2 B I 2.

ware ist ihm im Rahmen der Hardwareverkäufe gesichert. Demnach ist für einige wichtige Hardwarehersteller, die sich im Softwaresektor überwiegend mit der Herstellung und dem Vertrieb von Betriebs- und systemnaher Software befassen, der Softwareschutz tatsächlich uninteressant. Anderes gilt für die – wie es scheint – zunehmende Zahl von Hardwarehersteller, die auch im Bereiche der Anwendungssoftware und der sogenannten «turn-key-systems» tätig ist⁵³. Bei der Anwendungssoftware müssen die Hardwarehersteller mit den Softwarehäusern den Markt teilen. Anders als noch vor zehn Jahren können die Hardwarehersteller die Entwicklungskosten der Anwendungssoftware nicht mehr zu einem wesentlichen Teil über das Entgelt für die Hardware hereinbringen. Zum einen hat sich die gesonderte Berechnung der Software durchgesetzt⁵⁴ mit der Folge, dass sich die Hardwarehersteller mit den Softwareherstellern im Wettbewerb unter weitgehend identischen Voraussetzungen messen müssen. Zum andern hat sich das Wertverhältnis zwischen Hardware und Software derart stark verändert⁵⁵, dass der Software jetzt mehr als «nur» der Stellenwert eines Verkaufsargumentes zukommt. Zahlreiche Hardwarehersteller sind demzufolge wie Softwarehäuser auf eine gesicherte entgeltliche Veräusserung der Software angewiesen zur Deckung der hohen Entwicklungskosten und weil die Software wegen der Verschiebung der Wertverhältnisse als stark wachsender Anteil am Geschäftsvolumen zu einem unentbehrlichen Umsatz- und Ertragsfaktor geworden ist.

II. *Die Interessen der Allgemeinheit*

Nach überwiegender Meinung soll ein umfassender Softwareschutz mit den Allgemeininteressen gleichlaufen: Ansporn für die Kreativität, Zunahme der Investitionsfreudigkeit, vermehrte Bereitschaft zur Verbreitung, bessere Ausnützung der Ressourcen durch Vermeidung von Parallelentwicklungen und dergleichen sind die Argumente hierfür. Trotz des fehlenden oder unsicheren rechtlichen Schutzes stehen wir vor der Tatsache, dass sich dieser Wirtschaftszweig überaus stark entwickelt hat. Zu Recht bestehen verschiedenerseits Bedenken, dass ein zu grosser Schutzzumfang diese Branche ihrer Dynamik berauben könnte. Zu wenig klar hervorgehoben werden

53 NEUGEBAUER/DEHN/PHILIPP/WALTER, S. 159. Gemäss diesem Forschungsbericht ist in der BRD der Umsatzanteil der Hardwarehersteller an extern erstellter Anwendungssoftware (das ist die nicht vom Anwender erstellte «massgeschneiderte» und standardisierte Software) etwas über 50 %, wobei ihr Anteil an Standardsoftware wesentlich höher ist als derjenige der Softwarehäuser.

54 Vgl. betreffend Unbundling oben Kap. 3 A, Ziff. 3.

55 Oben Kap. 3 A, Ziff. 2.

rechtspolitische Bedenken gegenüber einem Softwareschutz, der auch inhaltliche bzw. den Problemlösungsideen zugehörige Elemente der Software umfasst:

Nach der Meinung zahlreicher Autoren⁵⁶ hätte der von ihnen befürwortete Softwareschutz nur einen praktischen Wert, wenn neben dem blossen Formschutz auch ganz oder teilweise der Algorithmus, sei dies nun der allgemeine Lösungsweg oder der für die Verwendung einer bestimmten EDV-Anlage konzipierte und programmtechnisch definierte Lösungsweg, quasi als Verfahrensschutz einbezogen werden könnte. Ein derartiger Schutz von Computersoftware müsste auf die Anerkennung eines patentähnlichen Rechtsschutzes für Problemlösungen nichttechnischer⁵⁷ Natur hinauslaufen. Damit müsste der Gesetzgeber in den Bereich der freien Problemlösungsmethoden oder/und Anweisungen an den menschlichen Geist eindringen, der in gewissen Fällen aus sozialetischen Gründen und im allgemeinen wegen der damit verbundenen praktischen, rechtstechnischen und beweisrechtlichen Probleme ausgespart werden müsste. Abgesehen von einer wenig wünschbaren weiteren Ausweitung des Rechtsstoffes, einer noch grösseren Komplizierung des Schutzes des geistigen Eigentums und der Bedenken gegen die rechtliche Sonderbehandlung⁵⁸ eines Wirtschaftszweiges, ist zu befürchten, dass ein derart weitreichender Rechtsschutz der Software zu untragbaren Einschränkungen der Handlungsfreiheit⁵⁹, zu Rechtsunsicherheit⁶⁰ und zu einer Abkehr von grundlegenden, herkömmlichen Rechtsprinzipien des geistigen Eigentums führen müsste.

56 Vgl. dazu u.a. ZAHN R., GRUR 1978, S. 208; KOLLE G., GRUR 1974, S. 10: Er spricht von «Lücke», weil das Urheberrecht den Algorithmus nicht erfassen kann; LEHMANN B., sysdata 5/78, S. 3: «Gegenstand des Programmschutzes sollten die geistig-schöpferischen Konzepte und Ideen sein.» KOKAI T. R., S. 26: «... benötigt jede umfassende Regelung des Software-Rechtsschutzes (...) Schutz der in einem Programm enthaltenen substantiellen Lösungen ...».

57 Zum überwiegend nicht-technischen Charakter der Computersoftware vgl. die patentrechtliche Praxis zum Schutz von Computerprogrammen unten Kap. 4 D.

58 Das Problem tritt stets dort auf, wo zu Sonderregelungen gegriffen wird, die nur sehr spezifischen Leistungen zugute kommen. Das aktuellste Beispiel für derartige Bedenken findet sich in der Diskussion über den Leistungsschutz der Interpreten, Ton- und Tonbildträgerhersteller und der Sendeunternehmen. Dazu PEDRAZZINI M., ZSR 96 (1977), S. 414: «Dass eine gewisse Willkür in der Auswahl der geschützten Leistungen liegt, ist nicht zu bestreiten – sie dürfte aber eine immanente Eigenschaft der Rechtsetzungspolitik sein, welche auf eine äussere Intensität der Interessen stark reagiert – was, vornehmer ausgedrückt, gelegentlich als organisches Wachstum der Gesetzgebung bezeichnet wird.»

59 SPOENDLIN K., ZSR 96 (1977), S. 425 mit Verweis auf die unten Ziff. 2 zusammengefasste Entgegnung auf die von PERRET F., ZSR 96 (1977), S. 205ff, entwickelten Vorstellungen für die Gewährung eines Rechtsschutzes für nicht-technische Problemlösungen.

60 Ebd.

1. *Erfordernis der Unterscheidung von freien und «computergerechten» Methoden zur Lösung eines nicht-technischen Problems*

Zu bedenken wäre im Zusammenhang mit einem weitergehenden besonderen Softwareschutz auch:

Der Computer (allgemeiner: der Automat) ist befähigt, anstelle des Menschen selbsttätig Vorgänge auszuführen. Der Gegenstand des menschlichen Handelns wird zu dem des Computers gemacht⁶¹. Es existiert eine Ebene, wo sich die Regeln oder die Methoden für das menschliche Handeln durch nichts von jenen des Automaten unterscheiden⁶². Gehören diese Regeln – wie es bei automatisierten, vor allem kommerziellen und wissenschaftlichen Aufgabenstellungen sehr häufig vorkommt – nicht zum Gebiet der Technik, so sind sie in der Urform als Anweisungen an den menschlichen Geist für die «konventionelle» Nutzung frei und würden bei einem Softwareschutz – sobald sie in einen maschinell ausführbaren Algorithmus konzipiert und aufgeschlüsselt sind – für die weit effektivere Nutzung mit Hilfe eines Automaten in dieser Aufschlüsselung monopolisiert. Die «computergerechte Veredelung» von freien Regeln oder Methoden (Tatbestand) würde Gegenstand eines beschränkten Exklusivrechtes für die Nutzung eines bestimmten Algorithmus (Rechtsfolge)⁶³.

Das bedeutet, dass die maschinelle Nutzung von freien Methoden oder Regeln teilweise oder vollständig zu Gunsten der Softwarehersteller monopolisiert würde, sobald die dafür – im wesentlichen Bereich und für die wichtigsten Programmiersprachen – vorhandene beschränkte Zahl von Algorithmen gefunden wäre. Ein derartiges Ergebnis des Softwareschutzes liesse sich schon mit Rücksicht auf den Erfinder der Urform einer untechnischen Problemlösung nicht rechtspolitisch bewältigen, wenn nicht gleichzeitig schlechthin der Schutz von untechnischen Methoden zur Diskussion gestellt würde. Ferner müsste die rein praktische Frage untersucht werden, wie ein Computeralgorithmus mit oft Tausenden von Elementen patentähnlich behandelt werden könnte. Es müssten die gleichen Bedenken auftreten, wie das früher in der Diskussion über den Patentschutz von Computerprogrammen der Fall war⁶⁴.

61 Beispiele: manuelle bzw. «konventionelle» oder automatisierte Buchhaltungsarbeit. Konventionelle und «computerisierte» medizinische Diagnosetechnik (vgl. unten Kap. 7 A.)

62 Oben Kap. 2 C I 1.

63 Dabei kann offengelassen werden, ob man unabhängige Parallelentwicklungen zulassen würde.

64 Vgl. zum Patentschutz unten Kap. 4 D.

Unüberwindlich erscheinen vor allem die Schwierigkeiten bei der Neuheitsermittlung. Es ist kaum vorstellbar, wie in einem Programm eines ganzen Programmpake-

2. *Zum Gegenstand des Rechtsschutzes von nicht-technischen Problemlösungen*

Der Schutz von EDV-Konzept und Algorithmus müsste nach einem allgemeinen Rechtsschutz für Problemlösungen nicht-technischer Natur ruhen. Mit SPOENDLIN⁶⁵ ist ein solches Ansinnen abzulehnen: Das entscheidende Problem stellt sich in der Beurteilung der Art der Handlungen, die der Schutzrechtsinhaber den Rechtsgenossen verbieten könnte. Beim (technischen) Patent ist der verbotene Bezirk klar abgesteckt, indem die Verwendung oder Inverkehrsetzung von bestimmten künstlich geschaffenen Gegenständen der materiellen Welt untersagt wird. Beim «nicht-technischen Patent» könnten wohl nur nach aussen gerichtete gewerbliche Anwendungshandlungen als Widerrechtlichkeitstatbestand in Betracht gezogen werden. Notwendigerweise müssten aber menschliche *Handlungen* beschränkt werden, die den «erfinderischen» Gedanken realisieren. Beschränkt werden könnten damit u.a.: Methoden zur Erfassung, Kontrolle, Bearbeitung und Ausgabe von Daten oder Informationen; Techniken, die von einem Computer bestimmungsgemäss Gebrauch machen, die Anwendung von neuen Programmierungs-, Parameterisierungs- oder Sprachmethoden.

Die Monopolisierung von menschlichem Handeln (Sprechen, Befehlen, Schreiben, Lernen usw.) ausserhalb der Technik, d.h. losgelöst von einem die Erfindung charakterisierenden materiellen Substrat, gibt vor allem aus rechtspolitischen Überlegungen zu schweren Bedenken Anlass. Spontaneität und Kreativität würden so z.B. bei einem EDV-Fachmann stark gehemmt, wenn er die im Laufe der Jahre in seinem Erfahrungsschatz angesammelten Regeln und Methoden für die Respektierung von Schutzrechten immer wieder «filtrieren» müsste.

Die genannten Bedenken zum Schutze von untechnischen Lösungsmethoden können nur unwesentlich reduziert werden, wenn man in Analogie zum Urheberrecht unabhängige Zweitschöpfungen zuliesse und nicht das «Auch-Machen» wie im Patentrecht verbieten würde⁶⁶. Auch bei einer solchen Lösung dürften – abgesehen von den Problemen einer vernünftigen Realisierung – die Nachteile bei weitem überwiegen. Der für die Weiterentwicklung wünschbare offene Austausch von Konzepten und Algorithmen im Studium, Schulung, Weiterbildung usw. würde durch eine derartige rechtliche Lösung stark beeinträchtigt. Die Kenntnisnahme von anderen Lösungen würde den Dritten immer gleichzeitig auch benachteiligen, weil sie

tes, das selbst für den oder die Autoren kaum im einzelnen überblickbar ist, Hunderte oder Tausende von erheblichen Algorithmusenlementen von ebensovielen unerheblichen auseinanderzuhalten wären und die Neuheitsfrage (oder die Prioritätsfrage) aufgrund einer Gegenüberstellung mit ebenso umfangreichen und komplexen zahlrei-

ihn seiner «Unabhängigkeit» und des Rechtes auf die Zweitschöpfung beraubt. Der «Befangene» könnte dadurch gezwungen werden, auch bei gleichen Lösungsvoraussetzungen (Programmiersprache, Computersystem usw.) wider besseres Wissen auf einen anderen, unter Umständen weniger optimalen Algorithmus auszuweichen. Das kann in einzelnen Bereichen einer Aufgabenstellung sehr behindernd und umständlich sein, wenn auch im Rahmen eines ganzen Programmes meistens «genügend» alternative Algorithmen zur Verfügung stehen.

Zusammenfassung

1. Die Bedeutung der Computersoftware als verselbständigtes und verkehrsfähiges wirtschaftliches Objekt ist bereits gross und noch stark zunehmend.

2. Der «Selbstschutz» und die Massnahmen zum Schutz der Software reduzieren das Verletzungsrisiko erheblich. Das gilt jedoch nicht in gleichem Masse für alle Programmtypen. Obwohl diese Massnahmen mit beträchtlichem Aufwand noch verfeinert werden können, steht das Erfordernis eines ergänzenden rechtlichen Schutzes ausser Frage, weil ein schwer einschätzbares Verletzungsrisiko übrigbleibt.

3. Vor allem im Bereich der Anwendungssoftware ist ein für Hardware- und Softwarehersteller in gleichem Masse erhebliches Schutzinteresse festzustellen. Mit Rücksicht auf die Allgemeininteressen (Freihalterfordernis, Praktikabilität, Rechtssicherheit) und aufgrund der grundlegenden herkömmlichen Bewertungen des Rechtsschutzes an geistigem Eigentum kann de lege ferenda ein weitreichender Schutzzumfang, der auch den Inhalt an konzeptionellen und algorithmischen Lösungen erfasst, nicht in Betracht gezogen werden. Ein rechtlicher Schutz dieser Elemente muss zum vornherein in den Bereich des Geheimnisschutzes verwiesen werden.

chen Gebilden entschieden werden könnte. Deutlich gegen einen patentähnlichen Konzeptschutz aus Praktikabilitätsüberlegungen (Kontrollaufwand, Unsicherheit, zu erwartende Verletzungshäufigkeit usw.) SCHRÖDER O., Referat vom 13.4.73 (Protokoll), S. 9: «Der vom Urheberrecht gebotene Schutzzumfang würde den Gegebenheiten daher besser entsprechen als der breite, vom Patentrecht (bzw. patentähnlichem System) gebotene Schutz».

65 Vgl. Zitat oben Abs. II. Siehe auch dort die ausführlicheren, im folgenden gekürzt wiedergegebenen Begründungen.

66 Ebenfalls skeptisch SPOENDLIN M., ZSR 96 (1977), S. 426.

4. Kapitel: Rechtliche Schutzmöglichkeiten der Software ausserhalb des Urheberrechtes und eines Sonderschutzrechtes de lege ferenda

A. Einleitung

Für den Schutz einer geistigen Leistung bestehen zwei Systeme: Einmal das System des direkten, unmittelbaren Schutzes durch Einräumung von Ausschliesslichkeitsrechten an bestimmten immateriellen Gütern. Die Rechtsordnung lässt an diesen immateriellen Gütern Rechte entstehen, verleiht diesen Gütern also Rechtsobjektcharakter. Für den Softwareschutz kommen nach diesem System das Patent- und das Urheberrecht in Betracht. Dem steht eine indirekte, mittelbare Schutzart gegenüber, bei der die Leistungen mit oder ohne Rechtsobjektcharakter in ein System von schützenden Verhaltensregeln eingebettet werden. Hierzu gehören Verhaltensregeln, die sich aus dem UWG ergeben und für alle Wettbewerbsteilnehmer Gültigkeit haben oder solche des Vertragsrechtes, die nur für den Vertragspartner verbindlich sind.

In aller Kürze haben wir auf die wesentlichsten Schutzmöglichkeiten¹ ausserhalb des Urheberrechtes einzugehen.

B. Der vertragsrechtliche Schutz

Die Möglichkeiten des unproblematischen und flexiblen vertraglichen Schutzes werden in der Praxis trotz der nur relativen Wirksamkeit voll ausgeschöpft². Regelmässig wird eine vertragliche Geheimhaltungspflicht³

1 Ausser Betracht bleiben die Vorschriften über unerlaubte Handlungen sowie Art. 162 des Strafgesetzbuches, weil sie als Verteidigungsmittel für Software-Interessen nur ausnahmsweise ins Spiel kommen können.

2 Siehe ausführlicher zum sog. Software-Lizenzvertrag LEHMANN B., sysdata 5/78, 8-9/78, 10/78; BRAUN B., BB 1971, S. 1344f; KOLLE G., GRUR 1973, S. 613; LUTZ H., GRUR 1976, S. 331ff; KOKAI T. R., S. 11ff; KLINTER W., ADL-Online 3/78, S. 174 inkl. Mustervertrag (S. 175ff), FUCHS A., S. 129ff und S. 492ff (Mustervertrag BVB-Überlassung vom 4. 11. 1977, Beilage zum Bundesanzeiger Nr. 216 vom 19. 11. 1977).

3 Damit wird gegenüber dem Interesse, die Software weiterzugeben, auf dem Wege der vertraglichen Bindung der Geheimnischarakter gewahrt. Der faktischen Ausschlusswirkung des UWG unterstehen Tatsachen, die weder offenkundig noch allgemein zugänglich sind, an deren Geheimhaltung der den Vorgang Beherrschende ein berechtigtes Interesse (Schutzinteresse) hat und die er tatsächlich geheimhalten will (subj. Erfordernis des Geheimhaltungswillens), BGE 80 IV 27 und StGB 162, BGE 88 II 322. Das Erfordernis des Geheimhaltungswillens wird durch die vertragliche Bindung der weiteren Geheimnisträger dokumentiert. Die Unkenntnis der Dritten darf nicht dem Zufall überlassen sein. Es genügt somit nicht, bloss darauf zu vertrauen, dass der unkundige Softwareerwerber im Normalfall die Software nicht «verstehen» kann

vereinbart. Sie vermag indirekt auch eine gewisse Schutzwirkung gegenüber Dritten zu entfalten, wenn ein bösgläubiger Konkurrent die Kenntnisse über den vertraglich gebundenen Softwareanwender erhält. Eine so ablaufende Kenntnisnahme kann mit den Verhaltensregeln des UWG geahndet werden⁴. Bedeutsam ist die vertragliche Bindung des Softwareerwerbers auch im Hinblick auf die dadurch erreichbaren präventiven Schutzmöglichkeiten. Im Vertrag können Modalitäten der Geheimhaltung von in der Software enthaltenen Geschäfts- und Fabrikationsgeheimnissen⁵, Art und Umfang des erlaubten Gebrauches, Vervielfältigungsmöglichkeiten, Missbrauchstatbestände, Sanktionen und insbesondere Kontrollrechte des Softwarelieferanten festgelegt werden. Nach LEHMANN steht heute «der Schutz der Programme durch vertragliche Abrede in der Praxis eindeutig im Vordergrund.»⁶ Es ist die einzige rechtliche Schutzform, deren Schutzzumfang und -wirkung zuverlässig abzuschätzen sind, weshalb stets auch dort, wo Urheberschutz und Patentschutz mit hoher Wahrscheinlichkeit bejaht werden können, eine vertragliche Flankendeckung sichergestellt werden muss.

C. Der wettbewerbsrechtliche Schutz⁷

Ein wettbewerbsrechtlicher Schutz ist ohne Zweifel gegeben, wenn die Software⁸ in Verbindung mit «besonderen Umständen»⁹ von einem Rechtssubjekt übernommen wird, welches mit dem Hersteller oder Lizenzgeber in einem Wettbewerbsverhältnis¹⁰ steht¹¹.

oder nicht befähigt ist, technische Massnahmen für die Weitergabe an Dritte zu veranlassen. Auch Arbeitnehmern und Beauftragten (Softwarehäuser und dergleichen) ist u.U. über die Dauer des Arbeits- bzw. Auftragsverhältnisses hinaus eine Geheimhaltungspflicht aufzuerlegen, soweit sich eine solche – vor allem bei höher qualifizierten Arbeitnehmern mit einer entsprechend hohen Stellung – nicht bereits aus den Umständen ergibt (BGE 64 II 172 und BGE 88 II 322ff).

4 Art. 1 Abs. 2 lit. f und g UWG; TROLLER A., IG II, S. 1073f.

5 Unten Kap. 4 C I.

6 LEHMANN B., sysdata 5/78, S. 4.

7 Ausführlicher zum wettbewerbsrechtlichen Schutz von wissenschaftlichen und technischen Leistungen SCHNEIDINGER C., S. 39ff; HELBER F., S. 36ff und die unten Kap. 4 C II zitierte Literatur.

8 Das gilt zivilrechtlich auch dann, wenn die Software ganz oder teilweise von einem Sonderschutz erfasst sein sollte. Zur Kumulation des Rechtsschutzes vgl. TROLLER A., IG I, S. 562ff.

9 Hierzu gehören primär Tatbestände nach Art. 1 Abs. 2 lit. a–k UWG, insbesondere Missbrauch von Geheimnissen, Vertragsbruch, Massnahmen, die eine Verwechslungsgefahr herbeiführen, Herabsetzung, Irreführung sowie allgemein Verstösse gegen Treu und Glauben, die nach sehr zurückhaltender schweizerischer Praxis in Anwendung der Generalklausel von Art. 1 Abs. 1 UWG erfasst werden. Vgl. u.a. TROLLER A., IG II, S. 1049ff; VON BÜREN B., Kommentar, S. 36ff; SIDLER P., S. 65f; HELBER F., S. 50ff.

10 Vgl. unten Abs. III.

11 Ausführlich SIDLER P., S. 50 und 54ff.

Im Vordergrund steht als unvollkommenes, faktisches Ausschlussrechtsrecht der Schutz des Geheimnisses¹². In allen Softwarebestandteilen können Tatsachen enthalten sein, die als Fabrikations- oder Geschäftsgeheimnis anzusehen sind¹³. Dabei handelt es sich vor allem um Geheimnisse, die eine Anleitung zum technischen und kaufmännischen Handeln vermitteln, einem Bereich von Gegenständen also, der a priori von einem Urheberschutz nicht erfasst werden kann. Geheimnisse aus dem Gebiet der Software können z.B. sein: Ungewohnte system- oder programmtechnische Verfahren, Tricks oder sonst einer grösseren Zahl von Konkurrenten nicht geläufige Lösungen, die eine besonders originelle oder rationelle Abwicklung eines EDV-Problems ermöglichen. Auch Techniken, die nicht zur Software gehören und nur bei Gelegenheit als sonstiges geheimes Wissen in die Programme eingingen, unterstehen dem Geheimnisschutz. Derartige Inhalte eines Programmes sind z.B. Fabrikationsverfahren, Rezepte, Abrechnungs-, Kalkulations-, Dispositions- und Bestellsysteme. Diese Geheimnisse müssen in ihrem Gutscharakter weder den Erfindungen entsprechen noch brauchen sie neu zu sein im Sinne des Patentgesetzes. Es genügt auch ein bescheidener, dem Fachmann naheliegender Fortschritt. Der Geheimnisschutz bietet in idealer Weise die Möglichkeit, die in der Software vereinigten heterogenen Wissensselemente ohne weitergehende Voraussetzungen in einen Schutz einzubeziehen. Er bildet vom Schutzgut her gesehen gewissermassen ein Komplement zum Urheberschutz. Er vermag den im wesentlichen auf die einfache Wiedergabe beschränkten urheberrechtlichen Formschutz¹⁴ dadurch zu verstärken, weil er ergänzend bei den dem Urheberschutz nicht zugänglichen inhaltlichen Elementen mit Verfahrenscharakter eingreifen kann.

Die Möglichkeiten des wettbewerbsrechtlichen Schutzes sind jedoch beschränkt. Etwelche Lücken und Unsicherheiten sind vorhanden.

I. Grenzen des Geheimnisschutzes

Der Geheimnisschutz ist nur solange gewährleistet, wie das Geheimnis besteht¹⁵. Der Geheimnisträger hat daher Massnahmen zu treffen, um das Bekanntwerden zu vermeiden. Das Geheimnis untersteht typischen Gefahren der Entwertung, die oft sogar vom Geheimnisträger selber ausgehen können. Nicht ausgewichen kann den von Dritten ausgehenden Bedro-

12 / Ausführlich BENDER D., Trade Secret Software Protection, S. 159ff.

13 Zum Begriff des Geheimnisses oben Kap. 4 B.

14 Kap. 6 im einzelnen. Der urheberrechtliche Schutz geht zwar weiter. Der einfache Wiedergabeschutz kann aber als eine «gesicherte Untergrenze» angesehen werden.

15 BGE 64 II 170.

hungen. Gelingen diesen Parallelentwicklungen, die nicht nur mitbenützt, sondern veröffentlicht werden, dann wird das Geheimnis zerstört. Die grössten Gefährdungen gehen aber vom Handeln des Geheimnisträgers aus. Vermeidbar sind Lücken im vertraglichen Schutz und Nachlässigkeiten bei der Geheimnisverwahrung (unsorgfältiges Verwahren von Informationsträgern). Unvermeidbar ist die Preisgabe von einzelnen Elementen des Geheimnisses bei der Marktbearbeitung, vor allem bei der Abgabe von Angebotsunterlagen. Bei der Software ergeben sich in der Praxis oft heikle Entscheidungen. Einerseits verlangt die Software als komplexer Leistungsgegenstand eingehende Erläuterungen, bis überhaupt etwas Wesentliches gesagt ist, andererseits ist angesichts der Vielzahl von potentiellen Nachfragern und der dafür erforderlichen breiten Marktbearbeitung eine sorgfältige Auswahl oder gar vertragliche Bindung der Geheimnisträger nicht immer möglich. Der zurückhaltenden Informationsbereitschaft des Geheimnisträgers steht das Bedürfnis des Erwerbers gegenüber, von möglichst vielen Seiten ausreichende Informationen zu erhalten, um die in der Software angewandten Lösungsprinzipien auf ihre Tauglichkeit zu prüfen. Obwohl in der Praxis gelegentlich derartige Schwierigkeiten zu überwinden sind, haben sich Handelspraktiken¹⁶ durchgesetzt, die bei den wichtigsten Geheimnissen keine Gefährdung mit sich bringen und dennoch die Verkehrsfähigkeit der Software nicht wesentlich beschränken. Der Geheimnisschutz bleibt in aller Regel bei dem meist begrenzten Kreis von Programmbenutzern intakt. Das ist nicht mehr ohne weiteres der Fall bei stark verbreiteten Softwareprodukten oder bei allgemeinem Marketing mit dem Geheimnis, wo selbst Konkurrenten oder ihnen nahestehende Firmen auf dem ordentlichen Weg in den Besitz des Geheimnisses gelangen können und damit von einem im Geheimnis vorhandenen schützenswerten Vorteil oder Vorsprung im Wettbewerb nicht mehr gesprochen werden kann.

II. *Das Wettbewerbsverhältnis*

Nach herrschender Ansicht¹⁷ weist der Begriff «Wettbewerb» auf das Bestehen eines (wirtschaftlichen) Wettbewerbsverhältnisses hin. Für die Anwendung der Bestimmungen des UWG ist stets ein derartiges Verhältnis erforderlich.

Eine auf die Übernahme einer Leistung hinauslaufende Handlung durch einen Dritten ist aus wettbewerbsrechtlicher Sicht nicht zu beanstan-

16 Vgl. oben Kap. 3 B 3.

17 BGE 90 II 54; 92 II 24; TROLLER A., IG II, S. 1035; VON BÜREN B., Kommentar, S. 19.

den, wenn dieser keine Vorkehrungen unternimmt, die übernommene Leistung auf den Markt zu bringen, und wenn der Übernehmer den Leistungsträger auch nicht durch den Gebrauch der Software als Mitbewerber derart berührt, dass dadurch seine Position im Markt verbessert und gleichzeitig jene des Erstleistenden beeinträchtigt wird. Sie sind nicht Mitbewerber, ihr Verhältnis fällt damit nicht in den Anwendungsbereich des UWG.

Der EDV-Anwender eines der Datenverarbeitung nicht zugehörigen Wirtschaftszweiges konkurrenziert den Softwarehersteller nicht, wenn er die unautorisierte Kopie strikte nur für den eigenen Gebrauch einsetzt. Der Usurpator ist in dieser, vor allem bei der allgemein verwendbaren Software (z.B. Anwendungen im Bereiche Rechnungswesen und Lagerwirtschaft) häufigen Situation nur Konsument, denn er trachtet nicht danach, als ein mit dem Softwarehersteller gleichgerichteter Wettbewerber mit bestimmten Partnern zum Abschluss zu kommen. Der Schutz bleibt aus, weil der Erstleistende durch das UWG nicht gegen den Ausfall von bestimmten Absatzmöglichkeiten geschützt wird, sondern nur dagegen, dass die unrechtmässig angeeignete Leistung nicht in Konkurrenz zur ursprünglichen treten und sich im Wettbewerb auswirken kann. Im UWG wird nur der Wettbewerb und nicht das allgemeine Verhalten im Wirtschaftsleben geregelt.

Von untergeordneter praktischer Bedeutung ist das UWG auch in einem anderen häufigen Fall, der die speziell für die Bedürfnisse eines bestimmten Wirtschaftszweiges eingerichtete Software betrifft. Wenn ein Unternehmen für eigene Zwecke spezielle Software erworben, weiterentwickelt und eingerichtet hat, ist es sehr unwahrscheinlich, dass es gewissermassen als Nebenerwerb anderen EDV-Anwendern die Software weitergibt. Ein solcher Erwerber müsste sich nämlich den gleichen EDV-Problemen gegenübergestellt sehen und mithin dem gleichen Wirtschaftszweig angehören. Wenige Unternehmen werden sich bereitfinden, den eigenen Konkurrenten Software weiterzugeben und derart Teile ihrer innerbetrieblichen Organisation, Geschäftsgeheimnisse oder Rationalisierungschancen preiszugeben. Selbst wenn die Weitergabe von speziell hergestellten oder erworbenen Programmen ausnahmsweise (bei Unternehmen mit ähnlichen EDV-Problemen und in bezug auf den Austausch von Software gleichgerichteten Interessen) im Interesse des Besitzers ist, so ist es wahrscheinlich, dass das Interesse an der Übernahme geschmälert ist, weil die Software allenfalls nicht mit den Aufgabenstellungen des Anwenders übereinstimmt oder Hindernisse bestehen unter dem Gesichtspunkt des technologischen Schutzes (vgl. oben Kap. 3 B).

III. *Unsicherheiten bei der Anwendung der Generalklausel des Art. 1 Abs. 1 UWG*

Ausserhalb der gesetzlich geregelten besonderen Tatbestände konzentriert sich die Diskussion¹⁸ immer wieder auf Handlungsgruppen, die sonderrechtlich nicht oder nur in unzureichendem Masse schützbar, wie es oft heisst: mit «Mühe und Kosten» verbundene Leistungen gefährden. Es handelt sich um die Nachahmung und die unmittelbare Übernahme¹⁹ einer Leistung.

Alle Softwarebestandteile, insbesondere die auf magnetischen Trägern aufgezeichneten, sind unter Vorbehalt der Massnahmen zur Datensicherung sehr einfach, schnell und billig zu reproduzieren²⁰. Sie sind durch Vervielfältigungstechniken gefährdet, die mit geringem Aufwand die Ausbeutung fremder Leistungen ohne nachschaffenden «Umweg» ermöglichen²¹.

Ohne auf die zahlreichen dogmatischen Auffassungen und die Abgrenzungsfragen zum wettbewerbs- und sonderrechtlichen Zuständigkeitsbereich einzugehen, bleibt im Ergebnis festzustellen:

1. *Im besonderen zur sklavischen Nachahmung*

Fehlt für ein Arbeitsergebnis der sonderrechtliche Schutz, so darf es nach überwiegender Auffassung²² nachgeahmt werden, falls nicht son-

18 Vgl. u.a. die zusammenfassenden Darstellungen aus deutscher Sicht bei WASSNER F., S. 79ff; HASS P., GRUR 1979, S. 361ff; aus schweizerischer Sicht bei TROLLER A., IG II, S. 1054ff; SCHNEIDINGER C., S. 39ff; PURTSCHERT J., S. 109ff; HELBER F., S. 56ff; SIDLER P., S. 65ff; PERRET F., ZSR 96 (1977) II, S. 326ff; Protokoll des Schweiz. Juristenvereins ZSR 96 (1977) II, S. 412f.

19 Bei der Nachahmung wird in mehr oder weniger enger Anlehnung der abstrakte Vorlagewert einer Leistung nachschaffend wiederholt. Keine gefestigte Auffassung besteht über das Wesen der «unmittelbaren» Leistungsübernahme. Die Abhängigkeit von einem fremden Leistungsergebnis ist nach überwiegender Auffassung stärker als bei der Nachahmung. Dabei wird «unmittelbar» umschrieben als «unveränderte Übernahme eines fremden fertigen Leistungsergebnisses» (BGHZ 44, S. 296 -Apfel-madonna), als «Ausnützung in unveränderter Form» (ebd.), als direkt beim corpus mechanicum, bei der körperlichen Substanz ansetzende Methode zur unveränderten Aneignung eines Leistungsergebnisses (eingehend PURTSCHERT J., S. 140f mit Hinweisen). Neben dem Sachverhalt der Unmittelbarkeit wird auf weitere, die Unlauterkeit mitbegründende Momente abgestellt. Insbesondere wird unterstellt, dass die Übernahme eines fremden wertvollen Leistungsergebnisses ohne eine ins Gewicht fallende zusätzliche eigene Leistung zur Förderung des eigenen Erwerbes und zu einer Schwächung der Wettbewerbsstellung desjenigen führt, der das Leistungsergebnis geschaffen hat.

20 Oben Kap. 2 B II 3 c und Kap. 3 B.

21 Dazu mit bes. Berücksichtigung der Computerprogramme SIDLER P., S. 69ff.

22 Siehe neben BLUM R., Diss. Zürich 1947, S. 21ff, auch die oben in Abs. III zitierte Literatur.

stige wettbewerbswidrige Begleitumstände mit der Art und Weise der Ausnutzung verbunden sind. Als verpönter Begleitumstand steht die durch die Nachahmung erreichte Verwechslungsgefahr im Vordergrund²³. Die Verwechslungsgefahr ist jedoch bei der Software sehr gering, weil die äusseren Gestaltungsmöglichkeiten sehr beschränkt sind, genauer, weil die Gestaltungsmöglichkeiten auf einen Bereich von zahlreichen, wenig kennzeichnungsfähigen Details beschränkt bleiben. Die unübersichtliche Vielzahl der die äussere Gestalt der Software (Druck-, Bildschirmbilder und dergleichen) bestimmenden Einheit von technisch-bedingten, lösungsmässig-funktionellen und gestalterischen Elementen vermag selten einen nachhaltigen Gesamteindruck zu bilden, der «beim Käufer eine Ideenverbindung mit einem bestimmten Hersteller oder einer bestimmten Qualitätsvorstellung erweckt²⁴.» Die als wesentlich erscheinenden Charakteristiken basieren auf vorbestehenden Methoden und Lösungsprinzipien (Buchhaltungssysteme usw.) oder sind durch die Hardware und die vorbestehende Software bedingt, weshalb ihre Kennzeichnungsfähigkeit nur sehr gering ist. Zudem kann der Ausstattungsschutz sich nicht auf angewandte Prinzipien, sondern nur auf die konkreten äusseren Gestaltungsmassnahmen beziehen. Auch liesse sich von einem Software-Nachahmer die Verwechslungsgefahr leicht vermeiden, weil ohne erheblichen Aufwand für die wichtigsten äusseren Merkmale Alternativlösungen realisiert, die verborgenen oder nicht kennzeichnungsfähigen zahlreicheren Elemente aber unverändert beibehalten werden können.

2. *Im besonderen zur unmittelbaren Leistungsübernahme*

Von erheblicher Bedeutung wäre es, darüber Klarheit gewinnen zu können, ob die unmittelbare Leistungsübernahme unlauterkeitsbegründend ist. Doch das Meinungsspektrum zur wettbewerbsrechtlichen Bewertung der unmittelbaren Leistungsübernahme ist breit²⁵. Die beiden Extrempositionen sind²⁶: UWG-Schutz als ein die Sonderrechte ergänzender subsidiärer Leistungsschutz²⁷ und UWG-Schutz nur als Schutz gegen die Begleitumstände der Ausnutzung, nicht aber als Schutz des fremden Arbeitsergebnisses und als Schutz gegen wirtschaftliche Nachteile, die eine derartige Ausnutzung

23 TROLLER A., IG II, S. 1055; Urteil des BGer in Praxis 66 (1977). S. 600ff oder Schw. Mitt. 1978, S. 59ff.

24 BGE 72 II 392, E. 3.

25 Siehe Darstellungen bei WASSNER F., S. 79ff; SCHNEIDINGER C., S. 39ff; PURTSCHERT J., S. 109ff.

26 Vgl. hierzu Ergebnisse von WASSNER F., S. 79ff.

27 HUBMANN H., Gewerbl. Rechtsschutz, S. 39f und S. 250ff, zitiert in WASSNER F., S. 79; de lege ferenda befürwortend auch PURTSCHERT J., S. 198.

für den Erstleistenden mit sich bringt. Im Gegensatz zu SIDLER²⁸ will SCHNEIDINGER²⁹ mit guten Gründen in tatbeständlicher Hinsicht keinen Unterschied zwischen der «sklavischen Nachahmung» und der «unmittelbaren Leistungsübernahme» anerkennen, weshalb beide zulässig seien. Nach TROLLER³⁰ und der wohl herrschenden Auffassung³¹ ist der Schutz gegen eine derartige Übernahme fremder Arbeitsergebnisse aufgrund der Generalklausel zu gewähren, weil «es verpönt ist, die Mittel eines andern sich dienstbar zu machen und sich aus seinem Vermögen, zu dem auch die Arbeitskraft gehört, unrechtmässig zu bereichern»³². Bei Zweifeln empfiehlt TROLLER³³, die Enumeration des UWG zu ergänzen. PEDRAZZINI³⁴ hat, obwohl er die eben erwähnte Auffassung von TROLLER grundsätzlich teilt, Bedenken, den Leistungsschutz im Rahmen des UWG generell zu erweitern, weil das eine ganze Reihe von Anspruchssuchenden aufweckt, für deren Leistungen das Schutzbedürfnis nicht nachgewiesen ist, ferner auch wegen der Schwierigkeit, aus der punktuellen und zurückhaltenden³⁵ Praxis³⁶ unserer Gerichte bei der Anwendung der UWG-Generalklausel sichere Prognosen ziehen zu können. Wie SCHNEIDINGER³⁷ zu Recht aus rechtssystematischer Sicht argumentiert, entspricht die Auffassung, dass eine Leistung des Schutzes vor krasser Ausbeutung bedarf, einem verständlichen Rechtsgefühl, das wohl im Gegensatz zu der rationalen Auffassung steht, dass Tatbestände, welche vom Gesetzgeber nicht sonderrechtlich erfasst wurden, im Interesse des ungehinderten Flusses der technischen und wirtschaftlichen Entwicklung für jedermann frei seien. Es geht nicht an, quasi mit dem UWG diesen Gesetzen eine zusätzliche Interessenabwägung über den Kreis der schützbaren Leistungen beizufügen, weil der Gesetzgeber mit der sonderrechtlichen Ausscheidung von einzelnen Erscheinungen bereits die Interessenbewertung abschliessend vorgenommen hat zwischen geistigen Erzeugnissen, die wegen ihres Eigenwertes in gemeinverträglicher Weise mit einem Monopol belohnt werden können, und solchen, die frei sind.

28 SIDLER P., S. 69.

29 SCHNEIDINGER C., S. 51 und dort zitierte Literatur.

30 TROLLER A., ZSR 96 II (1977), S. 429.

31 Vgl. Zusammenstellung bei SCHNEIDINGER C., S. 52.

32 TROLLER A., IG II, S. 1068; Siehe auch TROLLER A., Schw. Mitt. 64/65, S. 223ff.

33 TROLLER A., ZSR 96 II (1977), S. 429.

34 PEDRAZZINI M., ZSR 96 (1977), S. 414.

35 TROLLER A., Lehrbuch, S. 144; ders., IG II, S. 1054.

36 Vgl. BGE 85 II 431ff und BGE 87 II 320 mit Verbot des unerlaubten Nachpressens von Schallplatten; ferner TROLLER A., Schw. Mitt. 64/65, S. 221 betr. analoger Anwendung einer aus Art. 4 Abs. 2 URG gewonnenen Vorschrift mit wettbewerbsrechtlichem Charakter auch für technische Zeichnungen gemäss Art. 1 URG. Der Schutz wird nur gewährt gegen die unmittelbare Ausnützung des in der Darstellung verkörpertten Arbeitsergebnisses.

37 SCHNEIDINGER C., S. 53f.

Eine Analyse der gegensätzlichen Auffassungen ergibt, dass die Rolle der unmittelbaren Leistungsübernahme im Rahmen des Wettbewerbsrechtes nicht mit ausreichender Sicherheit beurteilt werden kann. De lege lata besteht im Wettbewerbsrecht eine unsichere Schutzgrundlage für den Fall, dass Computersoftware nachgeahmt oder kopiert wird und nicht gleichzeitig besondere unlautere Umstände gegeben sind.

Ergebnis

Obwohl das Wettbewerbsrecht geeignet ist, einen grossen Kreis von Übernahmehandlungen zu erfassen, verbleiben Lücken. Es stellt sich die Frage nach dem Schutz durch geistig-gewerbliche Sonderrechte. Der Geheimnisschutz bietet sich aber an als eine Ergänzung zum Schutze von Wissensselementen, die vom Urheberschutz a priori und vom Erfindungsschutz wegen der nicht-technischen Natur ebenfalls a priori auszuschliessen sind.

D. Der patentrechtliche Schutz³⁸

Bis Mitte der siebziger Jahre war die Frage der Patentfähigkeit von Computerprogrammen sehr kontrovers³⁹. Obwohl es nie an Befürwortern einer Anerkennung fehlte⁴⁰, lehnte es die schweizerische Rechtsprechung seit Beginn in konstanter Praxis ab, Computerprogramme zu patentieren, weil «ein solches Programm kein technisches Verfahren» darstellt, «da es nicht die Naturkräfte einsetzt»⁴¹. Programme wie auch die zugrundeliegenden Lösungsalgorithmen enthalten eine geistige Leistung, die lehrt, wie von einem vorhandenen technischen System (Computer) Gebrauch gemacht werden kann und wie die in ihm enthaltenen technischen Wirkungsmöglichkeiten zu koordinieren sind. Es handelt sich letztlich nur um Anweisungen, die sich an den menschlichen Geist bzw. an den Benutzer des im Automaten innewohnenden technischen Prinzips wenden. Naturkräfte gelangen nur mittelbar zur Anwendung.

Die Praxis des Bundesgerichtes fügt sich in die Rechtsentwicklung in den kontinentaleuropäischen Ländern ein, «die für einen Patentschutz von Algorithmen und Computerprogrammen kaum mehr Raum lässt»⁴². Trotz der etwas apodiktischen Ablehnung des Bundesgerichtes bestehen in

38 Vgl. Literaturübersicht oben Kap. I B.

39 Vgl. KOLLE G., GRUR 1974, S. 10 und GRUR 1977, S. 58.

40 Vgl. u.a. R. E. BLUM, NZZ Nr. 568 vom 6. 12. 1973, S. 35; ZIPSE E., GRUR 1973, S. 123ff und GRUR Int. 1973, S. 185; KUMM, GRUR 1969, S. 319f.

41 BGE 98 Ib 396, zitiert nach Übersetzung in GRUR Int. 1974, S. 227.

42 So KOLLE G., GRUR 1977, S. 74.

Anlehnung an die in der BGH-Entscheidung «Dispositionsprogramm»⁴³ gemachten Differenzierungen noch bescheidene Spielräume für eine patentrechtliche Erfassung von Computerprogrammen⁴⁴.

Einmal kann das Programm in den Schutzbereich fallen, wenn es zugleich erfinderische Züge eines von ihm gesteuerten technischen Vorgangs enthält⁴⁵. Ferner sollte auch ein mittelbarer Schutz eines Algorithmus möglich sein, wenn dieser in eine schaltungstechnische oder sonst in eine technische Ausformung eingebettet ist⁴⁶. Weist auch der Trend auf eine vermehrte hardwaremässige Realisierung von Programmvorschriften hin, insbesondere bei grundlegenden Funktionen des Betriebssystems⁴⁷, so stellt dies «nur in wenigen Fällen eine brauchbare Alternative zur flexibleren Programmierung eines Universalrechners dar und wird oft gegenüber dem Stand der Technik nicht patentierbar sein»⁴⁸. «Trotz der – aus der Sicht der Informatik – grundsätzlich bestehenden Austauschbarkeit von Hardware und Software, die zur Annahme auch einer patentrechtlichen Gleichwertigkeit verführt, sind Computerprogrammierung und Computer-Engineering nach Ausgangssituation, Arbeitsweise und verwendeten Mitteln»⁴⁹ zwei verschiedene Dinge. Die rechtliche Ungleichbehandlung von hardwaremässigen und softwaremässigen Programmlösungen lässt dennoch Bedenken aufkommen. Abgesehen von Fällen, wo die schaltungstechnische Lösung wegen ihrer Starrheit nicht brauchbar ist, kann vom Hardware-, nicht aber vom Softwarehersteller durch die oft freistehende Wahl des einen oder andern im Nutzeffekt sich identisch auswirkenden Verfahrens ein erheblicher Unterschied im rechtlichen Schutz bewirkt werden. Ein rechtspolitisch noch zu überdenkendes Ergebnis, bei dem sich abzeichnet, dass programmtechnische Lösungen auf Universalprozessoren gegenüber Lösungen auf «fix geschalteten» Prozessoren benachteiligt werden. Wenn auch in beiden Fällen der Algorithmus als solcher nicht Gegenstand der Erfindung ist, wird damit die eine Prozessortechnologie bevorzugt, weil bei deren Verwendung ein mittelbarer Schutz möglich wird.

43 BGH in GRUR 1977, S. 96, siehe auch BGH in GRUR 1978, S. 705 (Programmablauf).

44 Vgl. ausführliche Kommentierung von KOLLE G., GRUR 1977, S. 74ff.

45 Dies kommt vor allem in Frage bei Prozessrechnern mit analogen Bauelementen, die Fertigungs- und Verteilungsprozesse überwachen oder steuern.

46 WITTE A., Mikroprozessoren und Patentschutz, GRUR 1978, S. 511. Dieser Autor sieht die Möglichkeit eines patentrechtlichen Schutzes für Mikroprozessor-Erfindungen, bei denen Erfindungsgegenstand eine Vorrichtung mit einem Mikroprozessor ist, der Abläufe der Vorrichtung (Bsp. Kraftstoff-Injektion) steuert, wobei der Erfindungsgedanke bei an sich bekanntem Mikroprozessor im Ablaufprogramm des Mikroprozessors liegt.

47 Vgl. oben Kap. 2 B I 1 mit Beispiel.

48 KOLLE G., GRUR 1977, S. 74.

49 KOLLE G., GRUR 1977, S. 69.

Eine weitere Möglichkeit der patentrechtlichen Erfassung von Computerprogrammen wird nach der erwähnten Rechtsprechung gesehen, wenn darin eine neue, bisher nicht übliche und nicht naheliegende Betriebsvorschrift enthalten ist. Im Vordergrund stehen hierbei Betriebssystemprogramme (System- und Organisationsprogramme), die im Unterschied zu Anwenderprogrammen ausschliesslich dem Betrieb der EDV-Anlage dienen und wegen der gesteigerten Verwendungshäufigkeit in der Programmhierarchie höher stehen. Nach der nicht unumstrittenen Praxis⁵⁰ des schweizerischen Patentamtes sind solche Programme patentfähig. Wie KOLLE zutreffend argumentiert, kann bei einem programmierbaren Computer diese unübliche und neue Art und Weise der Steuerung kaum je denkbar sein, weil «jede Programmierung eines Computers sich immer im bestimmungsgemässen Gebrauch der Anlage bewegen muss, wie er durch deren Systemstruktur und -funktionen festgelegt und möglich ist»⁵¹. Das gilt auch für die der Hardware näher stehenden, nicht exakt von anderen Programmen abgrenzbaren Betriebssystemprogramme, die wohl unter Berücksichtigung technischer Gegebenheiten konzipiert werden, primär aber doch nur die vorhandenen Hardware-Funktionen mit gesteigerter Verwendungshäufigkeit organisieren, ohne dass neue technische Lösungsideen erforderlich sind oder neue, noch «schlummernde» technische Arbeits- und Wirkungsweisen geweckt würden.

Sind zweifelsohne noch nicht alle patentrechtlichen Probleme der Datenverarbeitung endgültig geklärt⁵², so steht für die weiteren Betrachtungen als gesicherte Praxis fest, dass Algorithmen und Computerprogramme im Zusammenhang mit Universalrechnern, die für eine beliebige Datenverarbeitungsaufgabe eingesetzt werden können, nicht oder nur in kaum vorstellbaren Ausnahmefällen patentfähig sind. Dieses Ergebnis berührt aber aus praktischer Sicht kaum negativ⁵³. Es bestehen wegen des weiten Schutzbereichs

50 Vgl. ISELIN F., S. 92 und Hinweis von FINSLER C., S. 17.

51 KOLLE G., GRUR 1977, S. 73.

52 Vgl. z.B. die etwas abweichende, «grosszügigere» US-Rechtsprechung, die einen Weg eingeschlagen hat, der «einerseits eine unangemessene Monopolisierung durch Patentierung von reinen Algorithmen verhindert, anderseits jedoch den Weg für einen Rechtsschutz von Vorrichtungen, die nach einem programmgesteuerten Verfahren arbeiten, offen lässt», WITTE A., GRUR 1978, S. 511. Nachtrag: BLUMENTHAL D. A./RITTER B. D., GRUR Int. 1980, S. 81ff.

53 Vgl. Zusammenfassung bei KOKAI T. R., S. 17, und LEHMANN B., sysdata 5/78, S. 3: Die Nachteile liegen vor allem in den mit Patenterteilungsverfahren verbundenen Auflagen. Insbesondere wird die Offenlegungspflicht nachteiliger bewertet, als der Vorteil eines mit Risiken und grossen Beweisproblemen behafteten Schutzanspruches. Ferner wird dem Patentverfahren die erforderliche Flexibilität und Praktikabilität abgesprochen mit Berücksichtigung der bekanntlich häufigen Programmänderungen, notwendigen Verbesserungen und der zu erwartenden Komplexität in den zu beurteilenden Gegenständen. In den meisten Fällen könnte der Patentschutz gar nicht eingreifen, weil die geforderte Erfindungshöhe nicht erreicht wird (vgl. Anm. unten Kap. 7 A sowie Anm. oben Kap. 3 C I und Kap. 3 C II b).

umfanges – anders als bei einem urheberrechtlichen Schutz – kaum überbrückbare Interessengegensätze der beteiligten Kreise.

Ergebnis

Jede der untersuchten Schutzmöglichkeiten weist grosse Einschränkungen auf:

- Das Vertragsrecht gewährt nur einen relativen Schutz.
- Das UWG greift nur bei besonderen unlauteren Verhaltensweisen oder Umständen ein. Einen ergänzenden wettbewerbsrechtlichen Leistungsschutz gibt es nicht.
- Das Patentrecht gewährt für den vorläufig kommerziell wichtigsten Bereich der programmierbaren Universalrechner nach feststehender Praxis keinen Schutz. Auch bei einer grundsätzlichen Zuständigkeit des Patentrechtes wäre nur ein kleiner Teil der Software patentfähig.

De lege lata bleiben nur die Schutzmöglichkeiten des Urheberrechtes.

5. Kapitel: Urheberrechtliche Grundlagen

A. Zum Vorgehen

Wie kaum ein anderer für den Urheberschutz relevanter Gegenstand hat die Computersoftware, insbesondere das Computerprogramm, in den letzten Jahren zu einer Vielzahl von Stellungnahmen herausfordern können. Bei keinem anderen Gegenstand konnten sich die verschiedenen Auffassungen vom Wesen des Urheberrechts derart klar profilieren. Die meisten namhaften Urheberrechtsdogmatiker haben sich an dieser Diskussion beteiligt. Obwohl sie jetzt über fünfzehn Jahre andauert, blieb vor allem in der Schweiz das Ergebnis völlig unklar. Nicht einmal ein minimaler Konsens über die Existenz eines sehr beschränkten Schutzes wurde erreicht. Wozu also noch eine weitere Abhandlung beifügen, die diese oder jene bereits umstrittene Auffassung mit weiteren Argumenten unterstützt, kaum aber einen entscheidenden Beitrag zur Überwindung der verunsichernden Uneinigkeit leisten kann?

Im folgenden soll versucht werden, eine Lösung «zwischen den Fronten» zu finden. Wie noch zu zeigen ist, scheint sich der Software- und vor allem der Programmschutz hauptsächlich an zwei Fragen zu entscheiden. Einmal die Frage, inwieweit die Freiheit von Erkenntnissen, Lehren oder Anweisungen¹ mit dem Schutz dieser Objekte zu vereinbaren ist, zum anderen, inwieweit diesen Erkenntnissen Bedeutung für die individuelle Eigenschaft eines Werkes zukommt. Jene Autoren, die bei Verstandeswerken dem Inhalt, d.h. dem ganzen Bereich der verkörperten geistigen Leistung in irgendeinem Umfang eine Individualitätsbegründende Funktion zusprechen, leiten bei Programmen den Schutz ausschliesslich aus inhaltlichen Elementen ab, weil bei den Programmen keine individuellen Elemente in der äusseren Erscheinungsform bekannt sind. Dem steht die wohl noch herrschende schweizerische Lehre² gegenüber, welche dem Inhalt der Werke jegliche Individualitätsbegründende Bedeutung abspricht. Sie lehnt daher den urheberrechtlichen Schutz von Programmen vollständig ab, weil in der allein massgebenden Form³ oder im Mitteilungsträger keine individuellen Charakteristiken nachzuweisen sind.

1 Im folgenden werden die Ausdrücke «Erkenntnisse», «Lehren», «Ideen», «Anweisungen», «Lösungsideen» und die oben in Kapitel 2 C II 2 a genannten synonym behandelt: Gemeint ist damit alles, was den Inhalt eines nach h.M. wissenschaftlich zu qualifizierenden Werkes darstellt, was als Inhalt dieser Werke zum Ausdruck gelangt.

2 Unten Kap. 5 B II 6.

3 Unten Kap. 5 II 6 a, b.

Diese Ausgangslage zeigt, dass ein Versuch, einen Konsens durch Überbrückung der gegensätzlichen Auffassungen über die Bedeutung des Inhaltes von Verstandeswerken zu erreichen, aussichtslos ist. Das wäre nur dann möglich, wenn mit einer weitgehend überwundenen, weil unpraktikablen Betrachtungsweise an der Unterscheidung von Form und Inhalt festgehalten wird. Falls es nämlich in derart angelegten Untersuchungen gelingt⁴, zunächst in einer auf den Bereich der Form angelegten Analyse individuelle Elemente nachzuweisen, so lassen sich praktisch alle massgebenden Auffassungen auf die Bejahung eines (im wesentlichen auf die reine Wiedergabe beschränkten) Formschutzes verpflichten. Die in der Form nachweisbaren Charakteristiken sollen demnach dem primären Ziel dieser Arbeit dienen, im Interesse einer minimalen Rechtssicherheit Lösungsansätze zu zeigen, die von einer möglichst grossen Zahl von verschiedenen Auffassungen bejaht werden. Erst in zweiter Linie interessiert ein möglichst vollständiges und «richtiges» Ergebnis, bei dem selbstverständlich auch individuelle Elemente aus dem Inhalt in die Betrachtung einzubeziehen sind.

Wegen des Umfanges des vorhandenen Schrifttums müssen sich die nachfolgenden Ausführungen auf eine Auswahl der wichtigsten – vor allem aus schweizerischer Sicht – umstrittenen Probleme und Meinungen konzentrieren⁵.

B. Die Grundlagen der umstrittenen urheberrechtlichen Problemfelder

Aufgrund der oben bereits vorgenommenen⁶ Unterteilung der Softwareergebnisse nach den jeweils verwendeten Ausdrucksmitteln ist der Kreis der zu behandelnden Werkarten abgegrenzt. Es handelt sich um Sprach-, insbesondere Schriftwerke, um wissenschaftlich-technische Darstellungen oder um daraus gebildete Mischformen. Die Homogenität dieser

4 Ganz unproblematisch ist dieses Unterfangen allerdings nicht, weil die exakte Unterscheidung von Mitteilungsform und mitgeteiltem wissenschaftlich-technischem Gehalt kaum durchführbar ist. Die Mitteilungsform ist bei Programmen besonders eng und kaum trennbar mit ihrer Aussage verbunden, weil ja jeder einzelne festgelegte Programmschritt zugleich eine bestimmte Aussage einschliesst und nur in sehr beschränktem Umfange andere Aussageformen diese bestimmte Aussage ebenfalls darstellen können. Wenn man aber die Betrachtung vom einzelnen Programmschritt löst und grössere Programmeinheiten ins Auge fasst, lässt sich auch bei Programmen eine zur Mitteilungsform gehörende Schicht erkennen, die bei veränderbarem sprachlichem Kleid ein und dieselbe Erkenntnis bzw. ein einziges Programm darstellen kann, das im Ergebnis identische oder äquivalente Verarbeitungswirkungen erzeugt. Näheres in Kap. 6 C I unten.

5 Ausführliche vergleichende Untersuchungen der Literatur finden sich u.a. bei ZAHN R., GRUR 1978, S. 207ff; KOLLE G., GRUR 1973, S. 611ff und GRUR 1974, S. 7ff; SCHWAIGER/KOCKLER, S. 353ff; MOSER M., GRUR 1967, S. 639ff und oben Kap. 1 B.

6 Vgl. oben Kap. 2 B II 3 a.

Werkarten lässt zu, dass vor der Einzelbetrachtung die für sie geltenden urheberrechtlichen Grundlagen gemeinsam vertieft werden.

I. Die Zuordnung zu bestimmten Werkkategorien

Wie die vergleichende Literaturuntersuchung von ZAHN⁷ zeigt, bestehen gelegentlich Probleme, die Ergebnisse der Softwareherstellung, insbesondere die Programme, bestimmten Werkkategorien im Sinne des Urheberrechtes zuzuordnen. Die wohl häufigste Zuordnung⁸ zu den wissenschaftlichen Schriftwerken⁹ bzw. zu den wissenschaftlich-technischen Darstellungen¹⁰ bereitet auch im schweizerischen Recht einige Mühe.

1. Die Zuordnung der umgangssprachlichen Dokumentationen und kunstsprachlichen Programme¹¹

Die wissenschaftlichen Werke sind in Art. 1 Abs. 2 URG in einer beispielhaften, nach herrschender Auffassung nicht abschliessenden Aufzählung dem Oberbegriff «literarische Werke» zugeordnet, der im VE II Art. 1 Abs. 2 URG durch den Ausdruck «Schriftwerke» ersetzt wird¹². Diese Zuordnung besteht in Übereinstimmung mit der bisherigen Lehre und Rechtsprechung, die ebenfalls eindrücklich auf die literarische und nicht auf die wissenschaftlich-inhaltliche Seite abstellt¹³. Aus der Sicht des Ausdrucksmittels lassen sich die obgenannten Erscheinungen ohne weiteres als Sprachwerke ansehen.

Unklar ist, inwieweit aufgrund des gesetzlichen Attributs «wissenschaftlich» auch noch bestimmte inhaltliche Abgrenzungen zu berücksichtigen sind, denn es dürfte dem allgemeinen Sprachgebrauch gelegentlich wi-

7 ZAHN R., GRUR 1978, S. 207.

8 U.a. KÖHLER R., S. 40ff und ders., BB 1969, S. 1114 (mit ausführlicher Begründung für die BRD); FROMM/NORDEMANN, Kommentar zum Urhebergesetz, § 2, Anm. 16; im Zusammenhang mit wissenschaftlichen Werken ULMER E., Wiss. Werke, S. 10ff; TROLLER A., Ontologie, S. 414ff; SIDLER P., S. 26; AXTER A. und O., BB 1967, S. 609; MÖHRING PH., GRUR 1967, S. 273.

9 § 2 Abs. 1 Ziff. 7 UhrG, Art. 1 Abs. 2 URG.

10 § 2 Abs. 1 Ziff. 1 UhrG, Art. 1 Abs. 2 URG.

11 Vgl. im einzelnen die oben in Kap. 2 E I und III aufgeführten Softwarebestandteile.

12 Im Gegensatz zum WUA und neueren ausländischen Urheberrechtsordnungen, die die wissenschaftlichen Werke selbständig neben die literarischen und sonstigen künstlerischen Werke stellen.

13 Vgl. u.a. BGE 88 IV 123ff, 126; BGE 85 II 123; BGE 59 II 405. Damit will nicht gesagt sein, dass das BGr nicht wissenschaftlich-inhaltliche Elemente in die urheberrechtl. Würdigung einbezieht, sondern dass die Gesamtheit der einer literarischen Erscheinungsform innewohnenden individuellen Elemente über den urheberrechtlichen Schutz entscheidet, ungeachtet dessen, ob der Inhalt des Werkes sich zusätzlich noch durch diese oder jene wissenschaftlichen Qualitäten auszeichnet.

dersprechen, Ergebnisse der Softwareherstellung darunter einzuordnen. Die Einreihung der wissenschaftlichen Werke bei den Schriftwerken in Verbindung mit den von der Rechtsprechung ebenfalls geschützten Werken von geringerer geistiger Bedeutung («kleine Münze») lehrt, dass dieser Begriff nicht von jenen hohen Bezirken des wissenschaftlichen Strebens, die der gängige Sprachgebrauch als wissenschaftlich bezeichnet, abgrenzen will, sondern alle Darstellungen meint, deren Gegenstand Wissen oder Wissensvermittlung ist, wenn nur die darin enthaltenen Ergebnisse geistigen Schaffens von individueller Prägung sind¹⁴. Das Gesetz umgrenzt mit diesem Attribut nicht bestimmte schützbare Sparten geistigen Schaffens, sondern ordnet nur beispielhaft einen gewissen Bereich menschlicher Betätigung zu, der sich in urheberrechtlich-relevanter Weise als Literatur im weitesten Sinne¹⁵ artikulieren kann. Dem Begriff «wissenschaftliches Werk» wird im Gesetz keine inhaltlich bestimmte Autonomie zugewiesen. Es kann somit dahingestellt bleiben, ob die Ergebnisse der Softwareentwicklung dem Bereich «Wissenschaft» zuzuordnen sind oder nicht. Ausserdem kann davon ausgegangen werden, dass der im Urheberrecht in seinem weitesten Sinne gebrauchte, gegenüber dem normalen Sprachgebrauch stark überdehnte Begriff diese Ergebnisse sicherlich umfasst.

Eine andere Frage ist, inwieweit die in Lehre und Rechtsprechung für wissenschaftliche Werke entwickelten Grundsätze¹⁶ analog angewendet werden können. In der Tat sprechen viele Gründe dafür: Gleich wie bei wissenschaftlichen Werken haben wir bei der Software im Bereiche des Inhaltes wie auch im Verhältnis zwischen Inhalt und Form zahlreiche sachliche oder lösungsmässige Bedingtheiten¹⁷; ebenso besteht ein Primat des Inhaltes bzw. ein Schwerpunkt auf der angemessenen Gestaltung von systematisch-methodisch geordneten Erkenntnissen. Der Formgebung kommt nur eine (bei der Software allerdings sehr aufwendige) dienende Mittel-Funktion für die Nutzung der Erkenntnisse zu; das Wesen des Werkes liegt nicht in seinem formgegebenen, statischen Dasein, sondern im Gebrauch der darin enthaltenen Regeln zur Lösung eines Problems. Ebenso wie bei wissenschaftli-

14 Vgl. SIEGWART H., S. 47; TRÜEB H., S. 61. Nicht zu folgen ist hingegen den von TRÜEB H., S. 70/71 angestellten Überlegungen im Zusammenhang mit Logarithmenpapieren. Die Unterscheidung von schützbaeren Werken der Wissensvermittlung und unschützbaeren der praktischen Wissensanwendung (reine Erkenntnisverwendung) ist nicht von Bedeutung, weil die Schützbarkeit des Werkes nicht von dessen Zweck abhängt (vgl. Zitate unten, Kap. 6 F I).

15 TROLLER A., IG I, S. 408.

16 Ein besonderes Autorenrecht für wissenschaftliche Werke existiert nicht. Es gelten die allgemeinen Grundsätze des Urheberrechts (TRÜEB H., S. 71). Es geht um die von Lehre und Rechtsprechung entwickelten Regeln, die bei dieser speziellen Kategorie von Werken mit den allgemeinen Grundsätzen in Einklang zu bringen sind.

17 Vgl. SIEGWART H., S. 31.

chen Ergebnissen stellt sich ferner die Frage, wie weit mit Rücksicht auf die Interessen der Allgemeinheit die in der Software enthaltenen Lösungen freizuhalten sind.

2. *Die Zuordnung von zeichnerischen, schematischen und tabellarischen Darstellungen mit sprachlichen Elementen*

Eine erfreuliche redaktionelle Klarstellung bringt der VE II zum URG in Art. 1 Abs. 2, der die zeichnerischen, schematischen und tabellarischen Darstellungen dem Begriff «graphische Werke» eindeutig zuordnen lässt. Im geltenden Recht sind sie in Art. 1 Abs. 2 URG am ehesten bei den «sonstigen bildlichen Darstellungen wissenschaftlicher und technischer Natur» einzureihen. Unbeachtlich ist die «unglückliche» gesetzliche Systematik, die diese Darstellungen dem Oberbegriff «literarische Werke» unterordnet¹⁸. Für das Attribut «wissenschaftlich» gelten die Ausführungen unter Ziff. 1 oben.

II. *Der urheberrechtliche Schutz von wissenschaftlichen Werken*

Unstreitig ist, dass die Individualität¹⁹ wissenschaftlicher Werke aus der äusseren oder inneren Form²⁰ – der am Ausdrucksmittel orientierten Gestaltung einerseits, der strukturellen Ordnung eines Werkes andererseits – resultieren kann²¹. Ebenso ist allgemein anerkannt, dass ausserhalb des Bereiches der wissenschaftlichen Werke sich der Werkcharakter aus der Individualität des Inhaltes ergeben kann und auch inhaltliche Elemente dem Schutz unterstehen²². Meinungsverschiedenheiten ergeben sich erst bei den Werken der Wissenschaft, insbesondere bei Schriften oder Darstellungen wissenschaftlicher oder technischer Art, wie Pläne, Tabellen, Zeichnungen, Schemata u. dergl. Der für diese Werke ebenfalls anerkannte Grundsatz, dass die darin enthaltenen Gedanken, Erkenntnisse, Theorien oder Lehren

18 Vgl. TROLLER A., Schw. Mitt. 1965, S. 220 und SJZ 1964, S. 370f; SIEGWART H., S. 38.

19 Auf den Begriff «Individualität» ist nicht einzutreten. TROLLER ist zuzustimmen, wenn er in Auseinandersetzung mit KUMMERS Lehre feststellt, die Begriffe «Originalität», «Individualität», «persönliche» oder «eigenpersönliche Schöpfung» besagten alle dasselbe (IG I, S. 466). Deutliche Unterschiede zwischen der Kummerschen «Einmaligkeit» und dem deutschen «Werk» sind ebenfalls nicht auszumachen (vgl. WASSNER F., S. 41ff, im Ergebnis ähnlich GIRTH P., S. 16ff). Allgemein zum Werkbegriff: HEIM E., Diss. Freiburg 1971, SCHENK U., Diss. Freiburg 1977. Festzustellen ist in dieser Arbeit lediglich, wo die Individualität vorhanden ist.

20 Eine eingehendere Auseinandersetzung mit TROLLERS Terminologie und Unterscheidungen drängt sich nicht auf. TROLLER hat sich grundsätzlich zu KUMMERS Lehre bekannt (TROLLER A., ZBJV 1968, S. 232ff). Siehe auch Ziff. 6 unten.

21 PLANDER H., S. 25.

22 U.a. ULMER E., Verlagsrecht, S. 105ff; ders., Wiss. Werke, S. 15; HUBMANN H., 4. Aufl., S. 32f; KUMMER M., Werk, S. 8ff; TROLLER A., IG I, S. 508f (mit terminologischer Differenz).

im Interesse der Allgemeinheit als Gegenstand geistiger Auseinandersetzung und für die freie Entwicklung der Technik offengehalten werden müssen, wird in unterschiedlicher Weise dogmatisch erfasst und führt in Grenzfällen zu etwas abweichenden Beurteilungen.

Die ältere, aus schweizerischer Sicht wohl noch herrschende Auffassung spricht den in einer wissenschaftlichen Arbeit enthaltenen Erkenntnissen weiterhin jede Bedeutung ab und geht davon aus, dass nur die Individualität der gewählten Form wissenschaftliche Arbeiten als «Werke» auszeichnet, also nur wegen der Eigenschaften der Form ein urheberrechtlicher Schutz bestehen könne²³. Bei wissenschaftlichen Werken bleibt der Urberschutz ein reiner Formschutz.

Im Gegensatz dazu steht die Auffassung von ULMER²⁴ sowie jene in der deutschen Rechtsprechung²⁵, in der Rechtsprechung des schweizerischen Bundesgerichtes²⁶ und in einem Teil der deutschen Lehre²⁷. Nach

23 SIEGWART H., S. 22, 54ff; TROLLER A., IG I, S. 412ff; ders., Ontologie, S. 385ff, S. 397ff; PERRET F., ZSR 96 n.F. II (1977), S. 272ff; VE II zum URG, Erläuterungen der Expertenkommission, S. 5: ... jedenfalls immer die Texte ... und nicht ihr wissenschaftlicher Inhalt, die urheberrechtlich geschützt sind; SIDLER P., S. 38ff; TRÜEB H., S. 67ff. KUMMER M., Werk, S. 9ff und 114ff, letzterer distanziert sich klar vom Gegensatzpaar «Form und Inhalt» und lehnt bei diesen Werken den Einbezug von irgendwelchen inhaltlichen Elementen ab, immerhin S. 111: «... wird zu beachten sein, dass freie, von jeder Zweckmässigkeitsüberlegung unbeeinflusste und damit nun eben für den Aufbau urheberrechtlicher Individualität geeignete Entscheidungen durch den zu behandelnden Gegenstand oder aus didaktischen Gründen auf beschränkten Spielraum eingeengt sein mögen. (...) Wohl aber kann eine Vielzahl solcher je für sich nicht volle Individualität erreichender Entscheidungen insgesamt doch in die Individualität hineinwachsen.»

24 ULMER E., Verlagsrecht, S. 105ff; ders., Wiss. Werke, S. 15f.

25 U.a. BGH in GRUR 1959, S. 251/252 – Einheitsfahrtschein; Rheinmetall-Borsig-Urteil II, BGH in GRUR 1956, S. 284: «Technische Konstruktionszeichnungen können jedenfalls dann Urheberschutz genießen, wenn die Art der Darstellung eigenpersönliche Prägung aufweist oder es sich um eine Zeichnung handelt, durch die der Konstrukteur erstmals einem schöpferischen Gedanken darstellerische Form verliehen hat.»

26 BGE 56 II 418 (Architekturpläne von reinen Nutzbauten); BGE 57 I 69 («zweckmässige» Reklamezeichnungen); BGE 58 II 298/99 (Architekturpläne); BGE 59 II 401 (Stadtplan); BGE 64 II 162 (Technische (MAAG-) Tabellen); BGE 88 IV 123 (Lehrgang für das Schreibmaschinenschreiben); BGE 100 II 171 (Architekturpläne für den Umbau eines gewerblichen Betriebes); BGE 101 II 105 (wissenschaftliche Abhandlung über psychologischen Test). In all diesen Fällen, in BGE 88 IV 127 mit ausdrücklichem Verweis auf ULMER, hat das BGer die Schutzfähigkeit der Werke nicht nur aufgrund der Form beurteilt, sondern es berücksichtigte und schützte in einem sehr beschränkten Rahmen (BGE 88 IV 127 ausdrücklich) ebenfalls die Originalität der geistigen Leistung (also auch inhaltliche Elemente). Einzelne Entscheide wurden in der schweizerischen Lehre kritisch aufgenommen (vgl. u.a. TROLLER A., IG I, S. 412; ders., Ontologie, S. 398; SIEGWART H., S. 54ff). Nach ULMER E., Wiss. Werke, S. 5, hat das BGer die einschlägigen Fragen behandelt, ohne «die Problematik wirklich zu klären.»

27 Zusammenstellung bei PLANDER H., S. 25ff. Zu beachten ist die Zwischenposition von HUBMANN H., 4. Aufl., S. 35: HUBMANN beharrt auf dem zusätzlichen Erfordernis der Individualität der Form. «Wegen der *mitzuberücksichtigenden* (Hervorh. vom Verfasser) wissenschaftlichen Leistung genügt es, dass die zur Verfügung stehenden Ausdrucksformen so viele Wahlmöglichkeiten offenlassen, dass für jeden der beschränkten Zahl von Wissenschaftlern, die diese Leistung erbringen können und wollen, eine

diesen Auffassungen kann das Individuelle im Inhalt oder in der Form oder in beidem liegen, nicht notwendig ist, dass es sich auf jeden Fall auch in der Form finden lässt. Nicht nur der Akt der Formgebung fällt also für den urheberrechtlichen Schutz in Betracht, sondern das «... literarische und künstlerische Schaffen im Ganzen»²⁸.» «Der Grund für den Urheberschutz liegt in dem Schutz der individuellen, geistigen Arbeit, die sich in dem Werk manifestiert, nicht in dem ästhetischen Genuss²⁹, den das Werk vermittelt³⁰.» «Auch Eingebungen der Phantasie, Sammlung oder Sichtung des Stoffes sowie die gedankliche Verarbeitung fallen ins Gewicht»³¹. An die Stelle der Unterscheidung zwischen Form und Inhalt³² hat die Trennung zwischen den individuellen und den gemeinfreien Zügen des Werkes zu treten. Es gilt demnach die Formel: Alle Elemente einer im konkreten Werk sich manifestierenden geistigen Leistung sind urheberrechtlich geschützt, sofern sie nicht dem Gemeingut entstammen oder im Interesse der Allgemeinheit als Gemeingut ausgesondert werden müssen. Die Grenze hat zu verlaufen zwischen dem vorgegebenen Gemeingut und den im Interesse der Allgemeinheit auszunehmenden schöpferischen Elementen einerseits und den schütz-
baren individuellen Elementen von Form und Inhalt andererseits.

Was liegt im schützbaeren inhaltlichen Bereich? ULMER beantwortet es so: «Nicht der Kern der wissenschaftlichen Werke, sondern es ist das «Gewebe», das den Kern umlagert, was wir urheberrechtlich schützen³³».

Ausdrucksform zur Verfügung steht.» HUBMANN ist es mit Recht nicht ganz wohl dabei, die Beurteilung unter Umständen allein bei den inhaltlichen Elementen anzusetzen, weil nach seiner Auffassung die wissenschaftlichen Erkenntnisse selten individuellen Anforderungen genügen (vgl. HUBMANN H., Der Rechtsschutz der Idee, UFITA Bd. 24, 1957, S. 7ff) oder dort, wo die Form nichts oder nicht viel Individuelles hergibt, eine Mitteilungsexklusivität (siehe unten Ziff. 5) entsteht, die eine freie geistige Auseinandersetzung mit dem wissenschaftlichen Gegenstand behindert. Nach HUBMANN müssen deshalb die schöpferischen, inhaltlichen Elemente mit einer «reduzierten Formindividualität» verbunden sein.

28 ULMER E., Verlagsrecht, S. 109.

29 In Abgrenzung gegenüber den Thesen TROLLERS und einem Teil der schweizerischen Lehre.

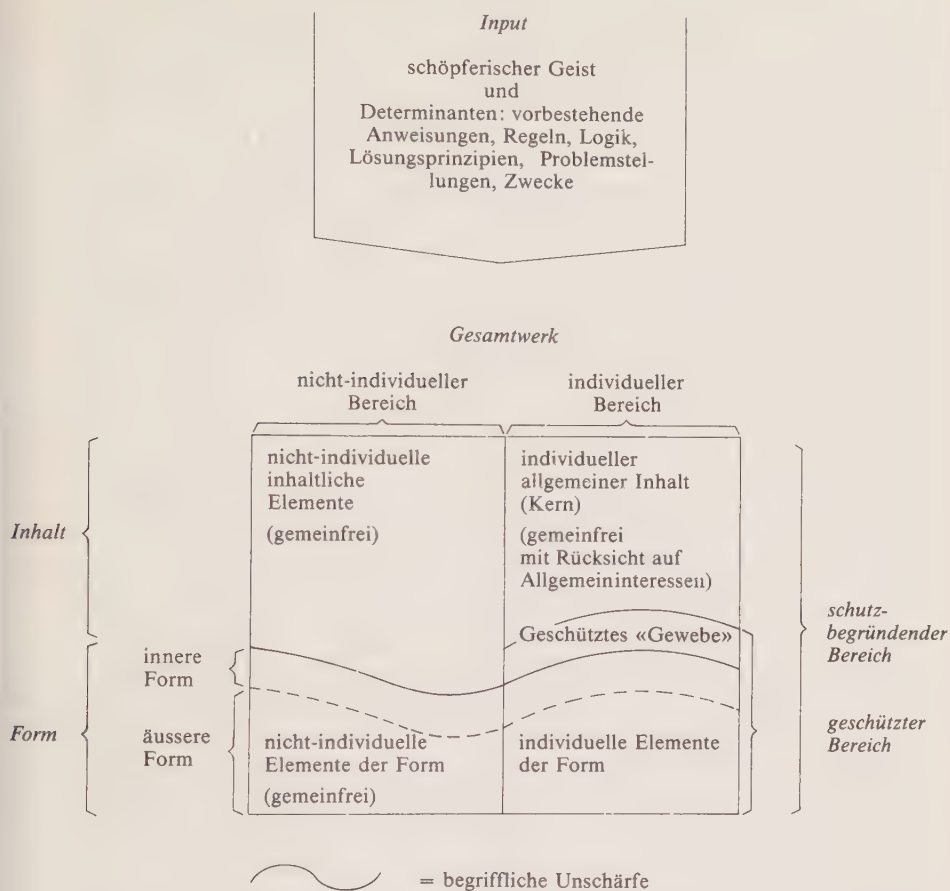
30 ULMER E., Wiss. Werke, S. 12

31 ULMER E., Verlagsrecht, S. 109; vgl. übereinstimmend BGE 88 IV 126ff; die Auswahl und das Erfassen des Stoffes, die Anordnung und Gliederung können gegebenenfalls geschützt sein.

32 Diese Unterscheidung ist noch ein im Schrifttum weit verbreiteter Ansatz. Doch TROLLER wendet sich gegen diese Unterscheidung, weil sie «nicht zureichend Auskunft gibt» (TROLLER A., Ontologie, S. 406). Jedoch bringt die von TROLLER angewendete Hartmannsche Wesensforschung des ästhetischen Werkes ausser pädagogisch wertvollen Erklärungen keine praktikable, neue und tiefere Einsichten. «Seine Schichtenlehre ist grundsätzlich nichts anderes als die Weiterführung der Aufgliederung in Form und Inhalt» (KUMMER M., Werk, S. 14). Offen bleibt, bis zu welcher Schicht, begonnen bei der äussersten, der Schutz reicht. Eigentliche ästhetische Qualitäten, wie oft missverstanden, verlangt TROLLER vom Werk jedoch nicht (TROLLER A., Ontologie, S. 405).

33 ULMER E., Wiss. Werke, S. 1.

Vorläufige Übersicht und Zusammenfassung



Den Begriff «Gewebe» übernimmt ULMER von GHIRON³⁴: «Die schöpferische Aktivität findet ihren Niederschlag in einem Komplex von Ideengehalten, die sich .. in den Aufbau (Gewebe) eines Buches ... und dergleichen verkörpern.»

ULMER präzisiert weiter den Spielraum der geistigen Leistung, welche seinem Kriterium der Schutzfähigkeit standhält: «... angesichts der Freiheit der einzelnen inhaltlichen Elemente» ist er zu sehen «in der Vielheit der

34 GHIRON M., Grundsätzliche Betrachtung über die Urheberrechte, UFITA 1932, S. 34ff, S. 38. Dieser Autor spricht auch «von einem Komplex von Ideengehalten, ... von einem inneren Zusammenhang eines wissenschaftlichen Werkes.»

Gesichtspunkte, in der Beziehung, in der sie zueinander stehen, und in der Art ihrer Darstellung ...»³⁵. Letztes Element ist als der Form zugehörend unbestritten. Der Inhalt wissenschaftlicher Werke ist also nicht schlechthin schutzfrei. ULMER³⁶ spricht von der Schutzzfähigkeit des Reichtums der Einfälle, der Fülle der Beispiele und Belege, der Vielfalt der gedanklichen Bezüge und der Lösungswege, die zusammen mit der Formgebung das Gewebe des Werkes ausmachen³⁷.

Einer Vertiefung bedürfen folgende Fragenkreise: Welche Bestandteile der Substanz genügen individuellen Erfordernissen? Wie verhält es sich, wenn nur individuelle Elemente der schützbaren oder nicht schützbaren Substanz gegeben sind und Individualitäten in der Form fehlen? Welches sind die Probleme, wenn nur im inhaltlichen, nicht aber im formellen Bereich Individualitäten nachzuweisen sind?

1. *Die Individualität der wissenschaftlichen Erkenntnisse (Inhalt)*

Neben den Freihalteerfordernissen mit Blick auf Allgemeininteressen besteht ein zweiter Ansatzpunkt für die Ausserachtlassung des Inhaltes geistiger Leistungen. Er besteht darin, dass bei Verstandeswerken dem Lösungs- oder Erkenntnisprozess, nicht aber dem Ergebnis selbst eine individuelle Qualität zugestanden wird. Die Ergebnisse seien – so ein Teil der Literatur³⁸ – entweder durch vorgegebene Naturgesetze, Anweisungen, Gesetze der Logik oder sonstige Determinanten im Schaffensvorgang (Zwangsläufigkeiten) bestimmt oder doch in ihrem individuellen Spielraum stark eingengt. Es kann nicht der Wille des Urhebers ein Ergebnis bestimmen, weil sich dieses aus der «Natur der Sache», als Vorgegebenes für jeden in identischer Weise oder nur mit geringen Abweichungen ergibt, wenn nur der notwendige Aufwand an Geist, Mühe und Geld in Kauf genommen wird. Zweckmässigkeitsüberlegungen und Vorgehensregeln sind, stets die richtigen Entscheidungen bestimmend, für den Schaffensverlauf kausal. Entsprechende Fähigkeiten vorausgesetzt, sind Objekte eines derartig verlaufenden Arbeitsprozesses potentiell jedermann zugänglich – und in diesem Sinne Gemeingut.

35 ULMER E., Verlagsrecht, S. 109.

36 ULMER E., Wiss. Werke, S. 15.

37 Gleiche Überlegungen in BGE 88 IV 123 (Lehrgang für Schreibmaschinenschreiben); Gleiches in der schweiz. Literatur z.B. auch bei TRÜEB H., Schw. Mitt. 58/59, S. 65f unter dem Gesichtspunkt der «inneren Form»: «... Kombination von Gedankengängen, in der Art, mit der sie zueinander in Beziehung gebracht werden, Akzentverlegung auf einzelne Schlüsse und Argumente» usw. Ebenso SIEGWART H., S. 28, der sich wie ULMER auf GHIRON M., UFITA 1932, S. 34ff, bezieht.

38 Vgl. die bei PLANDER H., S. 35ff, zusammengefassten Meinungen.

Aus solchen Überlegungen ergeben sich denn auch die wichtigsten Vorbehalte gegenüber einem urheberrechtlichen Schutz der Software. Ein engmaschiges Netz von vorgegebenen Bedingungen aus dem Bereiche der Problemstellung (Sphäre eines bestimmten Betriebes), der anwendbaren Lösungsmethoden (Lösungstechniken, vorgegebene Lösungsmodelle), der Maschine und ihrer vorbestehenden Softwarekomponenten (Sprachregeln u. dergl.) lässt die den Stoff fachgerecht verarbeitende Person in den Hintergrund treten. Diese besonderen lösungstechnischen Voraussetzungen bilden den «Perimeter» für die Lösung einer Aufgabe. Wie im einzelnen noch erläutert werden wird³⁹, zeigt die Erfahrung, dass auf den gleichen, stark vorgezeichneten Lösungsgrundlagen dennoch viele Lösungen entstehen können, die alle den gleichen oder, wie meistens gefordert, ähnlichen technischen und funktionellen Belangen zu genügen vermögen und mehr als eine routinemässige, handwerkliche Ausführung von Naheliegender darstellen. Es muss ja, um den Schutz des URG zu beanspruchen, nicht etwas absolut Neues («creation integrale»), eine ausschliesslich im schöpferischen Geist des Urhebers begründete Erkenntnis geschaffen werden. Der Urheberschutz kann bereits an eine relative und partielle Neuschöpfung anknüpfen. Diese kann darin bestehen, dass der Urheber Erkenntnisse eines Fachgebietes durch einen Aufwand an persönlicher geistiger Tätigkeit auf ein konkretes Problem anwendet und eine Lösung findet, die sowohl praktischen als auch ästhetischen Bedürfnissen zu genügen vermag⁴⁰. Den Schwerpunkt der geistigen Leistung bildet nicht wie bei künstlerischen Werken die Phantasie, welche im Unterbewusstsein wirkt und Erinnerungsbilder umformt und neue Bilder schafft, sondern die Sachkunde, welche sich aufgrund der Verbindung von Vorgegebenem und den vom Schöpfer dazugegebenen individuellen Anlagen und Fähigkeiten in der Entwicklung und Logik der Gedankenführung, in der Darstellung, Auswahl oder Anordnung manifestiert. Da nicht nur die künstlerische Phantasie das Tatbestandsmerkmal der Eigentümlichkeit begründen kann, sondern alles, was der Schöpfer dem Vorgegebenen mit seinen persönlichen Anlagen begeben kann, besteht grundsätzlich kein Anlass, wissenschaftliche Werke schutzlos ausgehen zu lassen. Belanglos ist allerdings entgegen der Auffassung des BGer⁴¹, ob der Inhalt eines Werkes von «hohem wirtschaftlichem und praktischem Wert» ist oder gar eine «Pionierarbeit» darstellt. Derartige Merkmale gehören in das Patentrecht.

39 Kap. 6 unten.

40 BGE 100 II 172, BGE 58 II 301f.

41 BGE 64 II 165f (MAAG-Tabellen).

Das Verhältnis des Schöpfers zum vorgegebenen Zweck und zu den Regeln der Computerbenutzung bedarf noch eingehender Klärung⁴³.

KUMMER engt den Zuständigkeitsbereich des Urheberrechtes in seiner auf einer unvollständigen Phänomenologie⁴⁴ beruhenden Untersuchung zu stark ein, wenn er «apodiktisch»⁴⁵ feststellt, dass «die Zweckfreiheit die durchgehende Eigenschaft des urheberrechtlich Schutzbaren sei»⁴⁶. Zur Begründung führt er an, die Zweckfreiheit (auch das «l'art pour l'art») erscheine als «elementare Regel», wenn man durchmustere, was bislang als Kunst aufgetreten sei. Dieses Kriterium führt ebenso zu einem unüberwindbaren Glaubensstreit wie andere Versuche, den Phänomenbezirk der Kunst gültig einzufangen. Weder bei Erscheinungen, die die Regel bestätigen, noch bei den zum Beweis angeführten Realien der Gegenwart lässt sich der Kunstcharakter verifizieren, «denn aufgrund der bruchlosen Entwicklung der Kunst (...) scheint es zweifelhaft zu sein, innerhalb dieses Bereichs eine Zäsur zu finden oder auf andere Weise eine Abgrenzung zwischen urheberrechtlich geschütztem und nicht schutzbarem künstlerischem Erzeugnis vollziehen zu können»⁴⁷.

Das Kriterium der Zweckfreiheit ist von KUMMER in anderer Hinsicht zutreffend begründet, wenn die mit dem Zweck verträgliche Freiheit als notwendige Voraussetzung für ein persönliches Schaffen angesehen wird. Wird mit gewissen Texten (wissenschaftlichen Berichten; politischen Reden usw.) ein Zweck verfolgt, so kann sich der Werkcharakter in ihnen nur im zweckuntergeordneten Spielraum zeigen. Zweckverträglich und frei handelt der Autor einer wissenschaftlichen Arbeit, wenn er «nach eigenem Gutdünken, unbeschadet des Inhalts seiner Aussage, ... ohne Nachteil für die Verständigung seine Worte so oder anders setzen kann»⁴⁸.

42 Aus diesem Gesichtspunkt ergibt sich das erste von zwei wichtigen Argumenten, die KUMMER (Werk, S. 202) gegen den Programmschutz ins Feld führt: Den Programmen zu wissenschaftlichen, technischen und verwandten (wohl auch kommerziellen) Aufgaben gebreche es, selbst im Kleinsten, an jedweder Zweckfreiheit. Das Programm sei ein Werk, dessen Systematik und Darstellung die wissenschaftliche Aussage aufs rigoroseste vorschreibt. Der Programmierer treffe jede der Entscheidungen so und nicht anders, «weil er sie als die zweckmässige oder eine von mehreren zweckmässigen halte» (Hervorh. vom Verfasser). Wo immer aber Zweckmässigkeit befehle, da bleibe der Urheberschutz aus. Auch in der Summe aller Entscheidungen trete er nicht ein, weil jede Einzelentscheidung, das Zweckmässige, das wissenschaftlich Richtige suchend, auch erste Ansätze einer Individualität nicht aufkommen liesse.

43 Die hierzu nachstehenden Ausführungen (Ziff. 2ff) gelten selbstverständlich für den Bereich der Form und des Inhaltes.

44 KUMMER M., Werk, S. 4.

45 SAMSON B., UFITA Bd. 56 (1970), S. 117, 112.

46 KUMMER M., Werk, S. 74; bestätigt ders., Entgrenzung, S. 91ff.

47 RAU G., Antikunst und Urheberrecht, UFITA Bd. 58 (1978), S. 28.

48 KUMMER M., Werk, S. 74.

Alle verfügbaren Wortlaute eines solchen Textes sind dem Zweck verpflichtet, eine bestimmte Aussage auszudrücken, also zweckgebunden. Ihre Zweckbindung neutralisiert sich bis zu einem bestimmten Grad durch die Anwesenheit von äquivalenten Alternativen, weil die Auswahl aus tatsächlich oder vermeintlich Gleichwertigem nicht einer sachlich bedingten Präferenz, sondern ausschliesslich dem freien Willen des Autors folgt. Die beschränkte Zahl an Alternativen begründet in einer einzelnen Entscheidung keine volle Individualität. Die zu einem Ganzen vereinigte Menge von Entscheidungen vermag aber die Individualität zu erreichen⁴⁹. Wie noch zu zeigen sein wird, kann KUMMER diese Erkenntnis im Zusammenhang mit Software nur eingeschränkt rechtlich fruchtbar machen, weil er nicht genügend berücksichtigt, dass beim Vorhandensein von Alternativen mit – aus der subjektiven Sicht und mit dem begrenzten Erkenntnisvermögen des Schöpfers – gleicher «Zweckintensität» (zweck-äquivalente Alternativen) urheberrechtlich Relevantes passiert⁵⁰. Gleichwertige Alternativen gewährleisten, dass ein paralleles Verhalten mit an Sicherheit grenzender Wahrscheinlichkeit nicht vorkommt⁵¹.

Im übrigen lässt sich nicht bei jedem Werk, das als Ganzes einer Zweckbestimmung unterliegt, der Schutz schlechthin verneinen. Auch ein zweckerfüllendes Werk bietet Spielraum für eine schöpferische Entfaltung. Der Zweck, dem ein Werk dient, ist nach einem anerkannten Grundsatz unbeachtlich⁵². Erforderlich ist nur, dass sich der Schöpfer von den durch den Zweck vorgegebenen Zwangsläufigkeiten zumindest derart lösen kann, dass er entweder Freiheiten in der Auswahl von Zweckerfüllendem hat oder in schöpferischer Weise neue zweckerfüllende geistige Objekte kombinieren kann. Schöpferisch sind die Gedanken bereits dann, wenn nicht ohne weiteres zugängliche, nicht zu erwartende Zweckmässigkeitsüberlegungen zum Tragen kommen und nicht erst, wenn überhaupt keine Zweckmässigkeitsüberlegungen angestellt werden.

In diesem Sinne, allerdings nicht sehr klar ausgesprochen, lautet

49 Vgl. die oft kritisierten, in der Sache aber zutreffenden Begriffe «zusammengesetzte Individualitäten», «Bruchteilsindividualitäten», «Individualitäten kraft Auswahl», KUMMER M., Werk, S. 42ff.

50 Ungenau deshalb auch KÖHLER R., S. 61: «Der Spielraum, der darin besteht, dass es oft mehrere richtige (äquivalente) Lösungswege gibt, muss daher ausser Betracht fallen.» Nicht nur ebenbürtige Elemente der Form, sondern auch solche der Lösung (Inhalt) können sich in der Zweckbindung auf eine gleiche Stufe stellen und eine freie Auswahl zulassen. Auch wenn der Auswahlentscheid u.U. inhaltlich nicht geschützt werden kann (z.B. als gemeinfreies Element einer neuen Lösung), trägt er doch als freie Entscheidung, als Ausdruck des freien Schöpferwillens zur Individualität (eventuell als Bruchteilsindividualität) bei. Die Auswahl darf aber im einzelnen nicht sachlich, d.h. lösungsmässig bedingten, eindeutigen Präferenzen folgen.

51 BENSE, Programmierung des Schönen, Allgemeine Texttheorie und Textästhetik, Baden-Baden 1960, S. 51.

52 TROLLER A., IG I, S. 417 und unten Kap. 6 F I.

auch die Rechtsprechung des BGer: Ein bestimmter Zweck ist mit der urheberrechtlichen Voraussetzung der Individualität zu vereinbaren⁵³, «wenn innerhalb des Zweckmässigen unter Umständen genügend Platz für individuelles Schaffen offengelassen ist»⁵⁴. Insbesondere schliesst ein Nützlichkeitszweck den Urheberrechtsschutz keineswegs aus⁵⁵.

3. *Die Anweisungen*⁵⁶

«Nach unbezweifeltem Grundsatz ist die Anweisung, die Handlungsregel, (das Rezept), die Manier, der Stil nie geschützt, vielmehr nur das bestimmte Werk wie hier in Wort und Strich vollzogen⁵⁷.» Wohl kann die individuelle Fixierung einer Anweisung als Schrift- oder Kunstwerk geschützt sein⁵⁸, der Inhalt der Anweisung und Ergebnisse ihrer Ausführung fallen ausserhalb des Schutzbereiches. Der Anweisungscharakter «ist eben eine absolute Sperre»⁵⁹ und schliesst jeglichen Urheberschutz aus. In letzter Konsequenz müssen bei diesem Grundsatz Zweifel aufkommen. Sind nicht die Manuskripte von schöngeistigen Reden, die Partituren von Symphonien, die magnetischen Aufzeichnungen von Bild und Ton, die Drehbücher von Filmen und schliesslich Computerprogramme in Form und Inhalt Anweisungen an ausführende Personen oder Geräte, Gegenstand einer Anweisungsexekution? Niemand wird bestreiten, dass die erwähnten Anweisungen – mit Ausnahme der letzten – in Form und Inhalt weitgehend schützbar sind, obwohl sich diese Werke in erster Linie und fast ausschliesslich als Anweisungen geben. KUMMER behandelt⁶⁰ das Problem so: «Eigenschaft eines

53 SIEGWART H., S. 35; KÖHLER R., S. 71 und dortige Zitate.

54 BGE 100 II 171: Umbaupläne, Zweckmässigkeit von Grundrissen bei der Inneneinrichtung eines gewerblichen Betriebes.

55 BGE 57 I 62 Erw. 3 und 4, BGE 56 II 417ff (Pläne von reinen Nutzbauten).

56 Es gehört sozusagen zum «Standard-Argument» der schweizerischen Lehre, dass Programme wegen ihres Anweisungscharakters nicht schützbar seien. Zusammenfassend HEIM/CURCHOD, S. 107f: «... dass die Programme aus einer Summe von Anweisungen (Befehlen) an den Computer bestünden und dass nach einem anerkannten Grundsatz Anweisungen immer schutzlos seien. Selbst wenn es sich um ungezählte und detaillierteste Anweisungen handle, die den vom Urheberrecht geforderten Grad an Individualität aufweisen würden, seien sie dessen ungeachtet nicht geschützt.» Vgl. auch PERRET F., ZSR 96 II (1977), S. 275f. Das Argument geht zurück auf KUMMER M., Werk, S. 210f und SIDLER P., S. 44ff.

57 KUMMER M., Werk, S. 52 (besonders eindringlich).

58 Zur Illustration TROLLER A., IG I, S. 458, 474: Beispiel Kochbuch–Kochrezepte.

59 KUMMER M., Werk, S. 202.

60 KUMMER M., Werk, S. 57: KUMMER hat das Problem gesehen und in Entgrenzung, S. 106, auch ein Beispiel geliefert: Es ist möglich, dass in räumlichen Koordinaten (Anweisungen) eine ganze (geschützte) Skulptur (Anweisungsexekution) definiert wird. Dieses Beispiel wird den nachfolgenden Erläuterungen zugrundegelegt. Man könnte aber ebensogut das Beispiel Partitur (Anweisung) und erklingendes Tongebilde (Anweisungsexekution) nehmen, wobei hier mit der herrschenden Lehre anzunehmen wäre, dass bei der Interpretation im Normalfall keine urheberrechtlich schützbaeren Beiträge mitspielen.

Werkes (hier Exekutionsergebnis, die Skulptur), solcherweise lückenlos in eine (nicht geschützte) Anweisung (Koordinatennetz) zurückversetzt zu werden, heisst nicht, auch das Werk (Skulptur) sei demzufolge nicht schützbar. Das ausgeführte Werk (Skulptur) will vielmehr auf seine Individualität hin untersucht werden. Wird dies bejaht, ist nun allerdings zugleich gesagt, dass also kein anderer auf ein gleiches Bündel inhaltsgleicher Anweisungen (Koordinaten) fallen könnte. Auch das ändert aber ... nichts an der Unmöglichkeit, den Inhalt der Anweisungen zu schützen, ungeachtet der Auswirkungen dieses Schutzes, dass jeder andere vom Gebrauch dieses (urheberrechtlich nicht geschützten) Bündels von Anweisungen ausgeschlossen wird.» KUMMER handelt mit seinem überdehnten Anweisungsbegriff und dem zu konsequent durchgeführten Grundsatz, alles, was als Anweisung vollzogen werden kann, aus dem Urheberrecht zu verbannen, nur eine Pattsituation ein, die er damit überwindet, auf der faktischen Ebene durch Ausschluss der Gebrauchsmöglichkeit dennoch Anweisungen zu schützen⁶¹.

Ein Teil des Wesens von Anweisungen bedarf der Erläuterung: Die Schutzunfähigkeit der Anweisung beruht darauf, dass Dritten nicht verwehrt werden soll, nach gleichen Rezepten Werke herzustellen, etwas «im gleichen Stil» zu machen⁶². Zwischen Anweisung und Exekutionsergebnissen bestehen in diesen Fällen Ähnlichkeiten, stets ist aber ein «gewisser Abstand» gewährleistet. Die Anweisungen führen nie zum schützbaren, konkreten Werk. Die Ergebnisse können sich verzweifelt ähneln, sich aber in concreto nie wie Vervielfältigung oder Bearbeitung zueinander verhalten⁶³. Die Ausführung vermittelt stets nur die Eindrücke eines Stils, einer Methode. Jede Person darf sich dieser Stilelemente bedienen. Dürfen wir aber ein Werk auch dann disqualifizieren, wenn zu seinem Bestand nur Anweisungen gehören, die – wenn von verschiedenen Personen ausgeführt – gleiches Werk sind und bleiben? Wenn in keinem Stil «ausgekleidet» werden kann, und wenn wegen der scharfen Konturen der Anweisungen der Einfluss des Anweisungsurhebers bis ins kleinste Detail bestimmend bleibt? Der Verfahrenscharakter der Anweisungen verflüchtigt sich dann praktisch vollständig, wenn unter allen Bedingungen, unter denen die Anweisungen eingesetzt werden, die Ergebnisse wie Vervielfältigungen oder Bearbeitungen wirken, wenn bei der Umsetzung der Anweisungen kein «schöpferisches Intermezzo» mehr Platz hat oder die Anweisungen und das damit erzeugte Ergebnis stets im Bereich des schützbaren, konkreten Werkes bleiben. Unter diesen Umständen sind die Anweisungen und die damit erzeugten Ergebnisse in sich geschlossen und

61 Vgl. auch das Beispiel bei KUMMER M., Werk, S. 206.

62 WASSNER F., S. 54. Vgl. auch DIETZ A., Urheberrechtsprobleme der neueren Kunstentwicklung, in: Film und Recht 1978 S. 90ff.

63 Siehe Beispiel von KUMMER M., Werk, S. 54: Geworfene Tintenkleckse usw.

berauben niemanden eines Rezeptes für das «Auch-Machen» mit anderen Spielarten. Die Beweggründe für die Freihaltung der Anweisungen fallen weg. Unter den nachfolgenden Bedingungen ist es deshalb durchaus angemessen⁶⁴, Erscheinungen mit Anweisungscharakter – sofern individuell – für die Beurteilung der Individualität heranzuziehen bzw. in den urheberrechtlich schützbaren Bereich aufzunehmen, sofern sie nicht mit Rücksicht auf Allgemeininteressen freizuhalten sind:

1. Wenn sowohl im Anweisungsträger (Koordinaten) als auch im Ergebnis (Skulptur) Elemente einer schöpferischen Leistung innewohnen, die in den Anwendungsbereich des Urheberrechts fallen. Der Anweisungsträger hat eine reine Mitteilungsfunktion für einen bereits vorhandenen urheberrechtlichen Tatbestand.

2. Wenn die Ergebnisse der Ausführung bei einer unbestimmten Anzahl von zur Ausführung befähigten Personen (Modellierer, Interpreten usw.) oder technischen Einrichtungen untereinander wie Vervielfältigungen oder abhängige Bearbeitungen wirken. Diese Bedingung sichert die Identität von Anweisung und Ausführungsergebnis.

Mit anderen Worten: Wenn ein Werk des Urheberrechtes sich in erster Linie oder ausschliesslich als Anweisung gibt (das eventuell mangels origineller Darstellungsmöglichkeiten in der Form nicht ohne weiteres einem Urheberschutz zugänglich ist, Koordinaten sind z.B. nur Zahlen), wenn sodann die Ausführung dieser Anweisungen zu einem dem Urheberrecht unterstehenden Ergebnis führt und wenn eine beliebige Zahl von Ausführungsergebnissen wie Vervielfältigungen oder Bearbeitungen erscheinen, dann ist der Inhalt der Anweisungen – reduziert um den gemeinfreien Bereich – zu schützen.

Übertragen auf das sich als Bündel von Anweisungen präsentierende Computerprogramm will das heissen: Entsteht bei einem mehrmaligen Vollzug der Anweisungen eines Programmes ein identisches⁶⁵, urheberrechtlich schützbare Ergebnis, so können die Anweisungen eines Programmes im Rahmen der Gemeinverträglichkeit ebenfalls einen urheberrechtlichen Schutz beanspruchen. Es stellt sich die Frage nach dem wahrnehmbaren Ergebnis: U.a. handelt es sich um die in origineller Form präsentierten Outputs (Bildschirm- und Druckdarstellungen usw.) mit Inhalten (klug zusammengestelltes Datenmaterial), die «einen ungewöhnlichen Grad geschäftlicher Erfahrung, Gewandtheit, Wirtschafts- oder Rechtskenntnis oder in der Form «eine eigentümliche, nicht von selbst gegebene Anordnung» zei-

64 Vgl. ähnliche Überlegungen bei WASSNER F., S. 53f.

65 Es ist evident, dass der mehrmalige Vollzug durch Mensch oder Maschine in diesem Fall zu gleichen Ergebnissen führt. Einflüsse durch eine beabsichtigte maschinelle Aleatorik seien dabei ausgeklammert.

gen⁶⁶.» Aber auch jene ausführbaren Anweisungen, die lediglich die Funktion haben, den Computer in der im Programm bereits konkretisierten, schützbar, individuellen Anweisungsstruktur (Aufbau von deskriptiven und imperativen Anweisungen) zu orientieren, können in gemeinverträglicher Weise in den Schutz einbezogen werden (Kap. 6 C 5). Das sind beispielsweise Verzweigungsbefehle, die auf die verschiedenen, z.B. aus Gründen der Übersichtlichkeit gebildeten Schleifen, Verästelungen und Unterprogramme hinweisen. Zur Vermeidung von Missverständnissen bedarf das Ergebnis einer Kommentierung.

a) *Zusammenfassende Erläuterung*

Bei der Beurteilung von Programmen und sonstigen Werken, die Bestandteile mit Anweisungscharakter haben, ist aufgrund der obigen Überlegungen wie folgt vorzugehen: Zuerst ist zu untersuchen, ob die Anweisungen in einer originellen Form festgelegt sind (origineller «Wortlaut» und Aufbau eines Programmes). Das wird, wie beim Programm noch zu zeigen sein wird, bereits zum urheberrechtlichen Schutz der Form führen.

Anschliessend sind die inhaltlichen Bestandteile, die Anweisungen, zu prüfen. Diese sind nach KUMMERS Lehre a priori wegen ihres Anweisungscharakters weder schutzbegründend noch schützbar, nach ULMERS Lehre – wenn individuell – schutzbegründend, aber wegen Gemeininteressen nicht schützbar. Zu Unrecht, denn KUMMER schliesst in zu konsequenter Weise die Anweisungen von der urheberrechtlichen Beurteilung aus, ULMER bzw. eine aufgrund seiner Lehre erschienene Arbeit zu diesem Thema⁶⁷ lässt ausser Betracht, dass ein Teil dieser Anweisungen auch als Inhalt ohne weiteres in gemeinverträglicher Weise geschützt werden kann und muss.

In einem nächsten Schritt ist also entsprechend den vorangegangenen Überlegungen die Qualität der Anweisungen zu untersuchen. Die Ergebnisse verschiedener Anweisungsausführungen sind nach den erwähnten Kriterien (Mitteilungsfunktion, Werkcharakter, Identität im Sinne einer Vielfältigung oder Bearbeitung) zu vergleichen. Stimmen die Ergebnisse überein und sind sie individuell, dann sind auch die Anweisungen «einmalig». Der Schutz dieser Anweisungen beeinträchtigt niemanden. Für einen Dritten sind diese Anweisungen ohnehin blockiert, weil deren Ausführung in den Bereich eines geschützten Werkes führen muss⁶⁸. Demnach können Anweisungen dieser Art – als Inhalt – in die urheberrechtliche Beurteilung einbezogen und müssen – soweit gemeinverträglich – geschützt werden. Die

66 Nach ULMER E., Verlagsrecht, S. 120 und dort zitierte Rechtsprechung.

67 KÖHLER R., Diss. München 1968.

68 KUMMERS Argumente für die Freiheit der Anweisungen fallen somit ausser Betracht. Im Ergebnis stimmt KUMMERS Lösung überein.

Ausscheidung hat anhand des sich – in Form und Inhalt – präsentierenden neuen oder bereits vorhandenen Ausführungsergebnisses zu geschehen. Knüpft man am allgemein anerkannten Grundsatz an, dass bei wissenschaftlichen Werken Individualitäten in der Form ohne Beeinträchtigung der Allgemeininteressen geschützt werden können, so dürfen deshalb auch jene Anweisungen eines Programmes geschützt werden, die im Verarbeitungsergebnis das Individuelle in der Form bestimmen (anweisen) bzw. bei der Programmstruktur im Bereich des bereits Schutzbaren zur Ausführung gelangen.

Beispiel: Ein Programm erzeugt ein in Form und/oder Inhalt individuelles Ergebnis (Druckbild⁶⁹, Bildschirmbild, Datensatz). Hierbei haben Einflüsse von seiten der Eingabedaten aus Gründen der Einfachheit ausser Betracht zu bleiben⁷⁰. Erste Frage: Ist der «Wortlaut» des Programmes individuell? Wie noch zu belegen sein wird: ja. Demzufolge ist das Programm in der konkreten Form geschützt. Zweite Frage: Sind Ergebnisse der Anweisungsexekution identisch? Unter Vorbehalt aleatorischer Wirkungen in der Form ist das ebenfalls zu bejahen. Dritte Frage: Weisen die Ergebnisse in der Form Individualitäten auf? Ja, weil die Listendarstellung z.B. originell ist. Folgerung: Neben dem «Wortlaut» des Programmes sind auch jene Anweisungen des Programmes geschützt, die Individualitäten in der Form bestimmen.

b) Bemerkungen zum Ergebnis

Im Ergebnis ergibt sich aus den vorgelegten Thesen nicht sehr viel Neues. Es ist zu bedenken, dass die geschützten Computeroutputs bei jedem Programmgebrauch ohnehin «verbreitet» werden, das sich ergebende Verbotungsrecht hat deshalb auch eine gewisse Sperrwirkung für den Programmgebrauch. Zwei Vorteile dürfen aber nicht übersehen werden. Zuerst ein praktischer: Das Ergebnis eines Computereinsatzes kann jetzt einfacher beurteilt werden, genauso wie das Ergebnis einer «manuellen Operation» (Formular, Tabelle usw.), welches in anderen Fällen direkt einer Beurteilung zugrunde gelegt wird. Es ist nicht einzusehen, weshalb Computer-Ergebnisse anders beurteilt werden sollen, weil sie mit dem «Makel» behaftet sind, dass die darin offenbarte schöpferische Leistung zuerst in Anweisun-

69 In concreto: Eine Liste mit Tabellen und einer gleichbleibenden Anordnung von Überschriften, Kommentaren (Umgangssprache) und einer Sammlung von Buchhaltungskennzahlen.

70 Der von KUMMER M., Werk, S. 205, beschriebene «Effekt», dass ein mit oder ohne Aleatorik arbeitendes Programm gegebenenfalls nichtgeschützte Eingabedaten zu geschützten Ausgabedaten umzuwandeln vermag, bleibe hier ausser Betracht. Ebenso sind aleatorische Wirkungen des Programmes auszuklammern, weil diese im formgebenden Bereich nur in der «Computerkunst» eine Rolle spielen.

gen zerlegt wurde, während bei konventionell erarbeiteten Ergebnissen solche Bedenken erst gar nicht aufkommen, weil die dort vorgenommenen gedanklichen Operationen und darstellenden Handlungen direkt in das schöpferische Ergebnis einfließen. Es ist auch angemessen, in gleichem Masse wie die individuellen Elemente der Form die sie vertretenden, formbestimmenden Anweisungen im Programm zu schützen, weil für diese Anweisungen die Gemeinfreiheit ebenfalls nicht gefordert werden muss. Zum zweiten Gesichtspunkt: Der Programmschutz verbessert sich eindeutig, weil eine unzulässige Übernahme in vielen Fällen⁷¹ zu einer aufwendigeren Bearbeitung (Änderung der gut sichtbaren Druck- und Bildschirmbilder, Programmstruktur etc.) zwingt und die blosser Übertragung in andere Programmiersprachen (Übersetzung) nicht mehr genügt.

Klarzustellen bleibt noch, dass Anweisungen, die sich auch im ausgeführten Ergebnis nur als Schablone erweisen, keinen individuellen Beitrag darstellen⁷².

4. *Individuelle Elemente in- und ausserhalb der schützbaren Substanz*⁷³

Unbestritten ist, dass individuelle Elemente innerhalb der schützbaren Substanz (des «Gewebes») dem Werk zugerechnet werden müssen.

71 Besondere Bedeutung gewinnt dieser Schutz dadurch, dass immer mehr Programme nach dem Dialogprinzip aufgebaut werden mit der Verwendung von Bildschirmen mit vielen Darstellungsmöglichkeiten und dem Austausch von umgangssprachlichen Informationen.

72 Nochmals ist auf das Problem der Entwicklungsstufen hinzuweisen (vgl. oben Kap. 2 C). Massgebend für die Individualität ist die Gesamtheit der sich in einem mittelbar gemachten Objekt offenbarenden, individuellen geistigen Leistung, die Summe der darin enthaltenen Individualitäten, aber auch nur diese (vgl. zur Insichverständlichkeit u.a. BGH in GRUR 1959, S. 251ff. (Einheitsfahrtschein); GRUR 1963, S. 633ff. (Rechenschieber)). Eine andere, hier nicht interessierende Frage ist die Zuteilung der Urheberschaft:

Bei der Beurteilung des Schutzes einer Konstruktionszeichnung interessiert nicht, ob gerade der Zeichner den entscheidenden Beitrag erbracht hat. Massgebend ist die in der Zeichnung von Ingenieuren, Technikern und Zeichnern geoffenbarte geistige Leistung, die vor der endgültigen Festlegung in verschiedenen Arbeitsphasen erarbeitet und in diversen Arbeitspapieren, Entwürfen usw. weitgehend vorgegeben sein kann. Gleiches gilt bei der Software und beim Programm. Der Beitrag des Programmierers (Kodierers) kann sich bei der Definition der formbestimmenden Anweisungen eines Computer-Outputs als eine reine Routinearbeit erweisen. Es interessiert indessen primär nur, ob die im Programm umgestaltete und geoffenbarte Leistung von Organisatoren (Konzept), Analytikern (u.a. Entwurf der Computeroutputs) und anderen Mitarbeitenden individuell ist, weil sie dann in allen Verbreitungsformen den Urheberschutz geniesst.

73 Nach einhelliger Auffassung wird «der urheberrechtliche Schutz schon gewährt, wenn bloss ein geringer Grad selbständiger Tätigkeit vorliegt» (BGE 88 IV 126; BGE 59 II 405; BGE 77 II 380, mit Verweisen). Eine «simple originalité» genügt, gefordert ist nicht eine «originalité marquée» (BGE 100 II 172). Angesprochen werden im folgenden vor allem Gesichtspunkte der gerichtlichen Durchsetzung.

Nicht klar ausgesprochen wird von einem Teil der deutschen Lehre, wie es sich mit den Individualitäten ausserhalb des schützbaren Bereichs des Gemeingutes verhält. Zutreffend dürfte sein, auch diesem Bereich schutzbe gründende Bedeutung beizumessen, wenn auch das Allgemeininteresse zwingt, den Schutz u.U. auf die einfache Wiedergabe zu beschränken⁷⁴. Dafür sprechen u.a. folgende Gründe:

1. Bei wissenschaftlichen Werken, wo naturgemäss die Spielräume in der Formgebung sehr eng sind⁷⁵, wäre der Rechtsschutz wesentlich effektiver. Der Wissenschaftler hätte sich nicht vor dem Richter auf das «Glücksspiel» einzulassen, die im Bereich der Form vorhandenen «schmalen» Individualitäten «für den Normalbetrachter handgreiflich⁷⁶ zu machen»⁷⁷, während der dem «Allgemeininteresse geopfert Bereich» diesen Beweis ohne weiteres zulässt.

Gewaltsame Interpretationen zur Begründung der Individualität wie in dem in Fn. 75 zitierten Beispiel werden entbehrlich. Der Wissenschaftler wird in praktikabler Weise geschützt, soweit es mit dem Grundsatz der Freiheit wissenschaftlicher Ideen zu vereinbaren ist. Vorbehalten bleibt das Problem der Mitteilungsexklusivität.

2. Im Extremfall, wenn sich in einem wissenschaftlichen Werk weder im Bereich der inneren und äusseren Form noch im Bereich des «Gewebes» Individualitäten in praktikabler Weise nachweisen liessen, müsste eine bedeutende wissenschaftliche Leistung völlig schutzlos bleiben. Auch in solchen Fällen ist ein Wiedergabeschutz aufgrund des individuellen «Kerns» nur billig. Auswirkungen einer Mitteilungsexklusivität müssten aber im einzelnen sorgfältig mit den Interessen der Allgemeinheit abgewogen werden.⁷⁸

74 Zum Problem des Mitteilungsmonopols nachfolgend Ziff. 5.

75 KUMMER M., Werk, S. 70. Unzumutbar, krampfhaft wirkend und der Rechtssicherheit nicht gerade zuträglich sind die Folgen der schweizerischen Lehre, die nicht über die «Schwelle» der Form hinauszutreten wagt. Dazu ein Beispiel von KUMMER M., Werk, S. 122ff, 124: «Offenbar weist nun aber jede derartige zeichnerische Darstellung, ob schon sie sich der üblichen Darstellungsmethoden bedient (Ansicht im Schnitt, Kennzeichnung bewegter und fester Teile mittels verschiedener Schraffur usw.) eine Vielzahl freier Entscheidungen des Zeichners auf. ... Aber nicht das ist entscheidend, ob um Zentimeter oder Millimeter variiert wird, sondern ob auch im *«Millimeterbereich»* beliebige Variationen möglich und anzutreffen sind. (Hervorhebung vom Verf.) Denn was handbreit variiert Kohlenstrich beim Kunstmaler, ist haarbreit streuender Reissfederstrich beim Techniker.» (Hervorhebung vom Verf.)

76 Die nachfolgenden Ausführungen sind insofern entbehrlich, weil in der Regel bei allen Softwarebestandteilen schon in der Form genügend Individualität nachzuweisen sind.

77 Ganz klar ausgesprochen von KOLLE G., GRUR 1974, S. 10, Fn 90: «... hindert die Feststellung, dass die freizuhaltenden Elemente eines Programmes jedermann zur Benutzung offenstehen, nicht, sie zur Bejahung der Individualität des Programmes heranzuziehen.»

78 Vgl. im Ergebnis unten Kap. 5 B II 5.

3. ULMERS Lehre ist auch im Bereich der wissenschaftlichen Werke mehr als nur eine Neuauflage der überwundenen Aufteilung in die äussere und innere Form. Die innere Form⁷⁹ und das bei wissenschaftlichen Werken geschützte inhaltliche «Gewebe» haben viele Berührungspunkte. Wollte man in beiden Fällen nach diesen Kriterien beurteilen, so wäre ein undeutlicher Rückgriff auf den Sinngehalt erforderlich. ULMERS ganzheitliche Erfassung der schöpferischen Leistung lässt – zumindest bei der Beurteilung des Schutzes – vermeiden, dass mit solch unpraktikablen Hilfsbegriffen operiert werden muss. Allerdings lassen sich auch mit dieser Lehre bei der Frage des Schutzzumfanges ernsthafte Abgrenzungsprobleme nicht vermeiden.

4. Entgegen der herrschenden schweizerischen Lehre, die bei wissenschaftlichen Werken wegen der Gefahr der Monopolisierung des gedanklichen Inhaltes an der Unterscheidung von Form und Inhalt festhält, halten wir es mit jener Auffassung, bei der der Bereich, aus dem die Individualität stammt, nicht deckungsgleich sein muss mit dem Bereich, auf den sich das Verbotungsrecht erstreckt. Der Inhalt begründet die Originalität, wird aber – soweit im Allgemeininteresse erforderlich – vom Schutzbereich ausgenommen.

Der gedankliche Inhalt bleibt frei. Nicht Postulate des Leistungsschutzes, sondern jene für einen wirksamen und rechtssicheren Schutz von wissenschaftlichen und technischen Werken werden damit erfüllt. Gegenüber der schweizerischen Lehre wird allerdings die Konzession gemacht, dass – soweit praktikabel – die Unterscheidung von Form und Inhalt durchgeführt wird, um einen minimalen Konsens zu erreichen.

5. *Das Problem der Mitteilungsexklusivität*

Wird die Auffassung vertreten, dass die Individualität eines wissenschaftlichen Werkes sich nicht nur aus dessen Formgebung, sondern sich auch oder nur aus dem als solchen ungeschützten Inhalt ergeben kann, so können die Voraussetzungen eines urheberrechtlichen Schutzes erfüllt sein, wenn im Ausdrucksmittel minimale oder keine Alternativen vorhanden sind. Diese allerdings sehr seltenen Fälle sind in Wissenschaftsbereichen möglich, die sich mit Formeln, Zeichnungen oder unpersönlichen sprachlichen Mitteln ausdrücken. Das Ausdrucksmittel gibt dem Autor keine Gestaltungsspielräume, er muss «die Sache für sich sprechen lassen». Nicht auszuschliessen ist bei kürzeren wissenschaftlichen Beiträgen, deren Aufbau durchgehend einem sachlichen Prinzip folgt, dass keine Individualitäten in

der inneren Form auszumachen sind. Wer den wissenschaftlichen Inhalt wiedergeben will, ist also genötigt, sich des gleichen Ausdrucksmittels zu bedienen. Abstand zu wahren ist unmöglich⁸⁰. Der Schutz bewirkt dann ein Mitteilungsmonopol.

Diese Exklusivität ergibt sich nicht deshalb, weil das Urheberrecht den Inhalt schützt, sondern weil der auf die einfache Wiedergabe beschränkte Schutz notwendigerweise bei jeder geistigen Auseinandersetzung mit den Erkenntnissen verletzt würde. Eine untragbare Konsequenz, weil sie den freien Austausch von wissenschaftlichen Informationen beeinträchtigt⁸¹. Eine verbreitete Auffassung lehnt deshalb generell die Werkeigenschaft ab, wenn sich eine Erkenntnis nicht auf verschiedene Weise mitteilen lässt⁸².

Es ist nun aber mehr als eine reine Systemfrage – wie PLANDER⁸³ eindrücklich darlegt –, ob man das Problem der Mitteilungsexklusivität über den Werkbegriff löst mit der von der schweizerischen Lehre geforderten Individualität der Form, welche einen vom Urheberrecht nicht behinderten Fluss der wissenschaftlichen Erkenntnisse sicherstellt, oder ob man bereits dem unoriginell dargestellten, individuellen wissenschaftlichen Inhalt die Werkeigenschaft zugesteht und erst später im Einzelfall die aus dem Freihalteerfordernis sich ergebenden Konsequenzen zieht. Abgesehen davon, dass die vorzuziehende letztere Lösung weitgehend von unpraktikablen Hilfsbegriffen (äussere und innere Form) und krampfhaften Bemühungen um den Nachweis einer individuellen Form zu entlasten vermag, ist sie auch geeignet, einen sichereren Schutz zu bieten. Neben der Verpflichtung zur Quellenangabe und weiteren, im vorliegenden Zusammenhang nicht interessierenden Verbesserungen, die sich nach der schweizerischen Lösung nur über mühsame Hilfskonstruktionen erreichen liessen, wäre mit dieser Lösung gewährleistet, dass die vom Wissenschaftler schöpferisch erarbeiteten, seine Individualität ausweisenden Arbeiten vor der Veröffentlichung (Bekanntmachung), in der Sphäre seines privaten Gebrauchs vom urheberrechtlichen Schutz erfasst wären (Erstveröffentlichungsrecht und Recht der ersten Inhaltsmitteilung, § 12 Abs. 1, 2 UrG⁸⁴). Soweit der kurze Exkurs mit Verweis auf die zitierte Literatur.

80 TROLLER A., *Ontologie*, S. 400f (besonders hervorgehoben).

81 Nicht in jedem Fall würde der Mitteilung ein verwirklichtes Objekt (Maschine, Haus usw.) gegenüberstehen, das als Grundlage für eine neue unabhängige Zeichnung dienen könnte.

82 SIEGWART H., S. 50; TROLLER A., *IG I*, S. 412ff; ders., *Ontologie*, S. 399ff.

83 PLANDER H., S. 57ff. Zur ausführlichen Begründung sei auf diese Abhandlung verwiesen.

84 In Abweichung von der herrschenden Auffassung müsste das Geheimhaltungsinteresse des Urhebers vor der Erstveröffentlichung Vorrang haben vor dem Allgemeininteresse, den gemeinfreien individuellen wissenschaftlichen Inhalt bereits in dieser

6. *Die Form*

a) *Das Ausdrucksmittel*

Der geistige Gehalt wird bei den hier zu untersuchenden Erscheinungen in Wörtern⁸⁵, Sätzen, Zeichen oder Zeichenkombinationen, Linien, zeichnerischen Symbolen u. dergl. kundgetan. Nicht nur die Auswahl und Kombination dieser Elemente, sondern auch deren räumliche Anordnung und Gliederung (Zwischenräume bei Tabellen und Formularen) sowie zeitliche Aufeinanderfolge sind Aspekte des Ausdrucksmittels. Auch Zahlen fallen als Ausdrucksmittel in Betracht, weil sie Gedankeninhalte ausdrücken, jedoch fällt bei ihnen, soweit sie ihrer Funktion nach nicht über die Mitteilung der ihr eigenen mathematischen Bedeutung hinausgehen, in der Form keine urheberrechtliche Bedeutung zu. Die Konkretisierung dieser Ausdrucksmittel ist genügend, wenn diese erst durch die Verwendung eines Gerätes wahrnehmbar gemacht werden können⁸⁶. Nicht einzutreten ist auf die vor allem im Anschluss an KUMMERS Lehre vieldiskutierte Frage, ob die «menschliche Gestaltung» ein Erfordernis sei⁸⁷. Bei all den hier zu beurteilenden Gebilden ist die entstehungsmässige Verbindung mit dem Menschen gegeben.

b) *Die äussere und die innere Form*⁸⁸

Einigermassen klar ist der Bereich der äusseren Form, des blossen Ausdrucksmittels ohne jeden Rückgriff auf den Sinngehalt. Diffus ist der eingebürgerte Hilfsbegriff der «inneren Form». Die innere Form eines Werkes besteht in der Ordnung der Gedanken (Gedankenfolge, Gedankenbewegung, inhaltliche Disposition, Ideengruppierung), in der ein bestimmter Inhalt dargeboten wird⁸⁹. Nicht dazu gehören diejenigen Gliederungen der Gedanken, die in den Sinngehalt eines Werkes übergreifen. Dazu gehören systematische Elemente einer Arbeit, die bereits eine inhaltliche Aussage in

Phase der freien geistigen Auseinandersetzung vorzubehalten. Unzulässig ist die erste Inhaltsmitteilung auf jeden Fall, soweit der Inhalt nicht von der urheberschutzfähigen Gestaltung gelöst wird (vgl. unten 7 B V 1).

85 Nicht nur Wörter einer Umgangssprache oder sonst einer Wortsprache gehören dazu, sondern auch sprachähnliche Elemente von anderen Kommunikationsmitteln (Blinden-, Geheim-, Code-, Stenografie- und Symbolschriften), die auf Konvention beruhen und in Begriffen der Sprache erfassbar sind (HUBMANN H., 4. Aufl., S. 90).

86 Vgl. oben Kap. 2 E IV, Kap. 6 F I und auch MARWITZ-MÖHRING, Kommentar, Berlin 1929, S. 16.

87 HEIM E., S. 49f.

88 Diese begrifflichen Klärungen sind erforderlich, weil nach einem Teil der schweizerischen Lehre ohne sie nicht auszukommen ist.

89 HUBMANN H., 4. Aufl., S. 31; eingehend TRÜEB H., Schw. Mitt. 58/59, S. 65.

sich einschliessen. Genauerer lässt sich auch nicht durch die von TROLLER vertretene Schichtenlehre gewinnen. Er lässt bei der Ausscheidung der Form (des ästhetischen, des unmittelbar sinnlich Wahrnehmbaren) die Frage offen, wie die Ausscheidung ohne Umweg über die verstandesmässige Wahrnehmung vor sich gehen soll. Zudem vermag seine Lehre keine sicheren und praktikablen Kriterien zu liefern für die Unterscheidung von verstandesmässiger und ästhetischer Wahrnehmung. Nach KUMMERS Lehre existiert der Begriff der «inneren Form» nicht mehr. KUMMER subtrahiert bei wissenschaftlichen Werken Elemente des Inhaltes (zu Recht) derart rigoros (genauso wie ULMER und HUBMANN), dass er irgendwo im Mittelfeld zwischen (geschützter) Darstellung und (nicht geschützter) inhaltlicher Aussage stecken bleibt⁹⁰.

Es bleibt festzuhalten, dass – unabhängig von den verschiedenen Blickwinkeln und Begriffen – im Ergebnis neben der äusseren Mitteilungsforn nur die frei variierbare, d.h. die nicht wissenschaftlich naheliegende und einzig richtige, rein sachbedingte Gliederung und Anordnung, Planung, Auswahl und Erfassung des Stoffes schützbar ist⁹¹.

Ergebnis

Bei der Abgrenzung des schutzfähigen Bereiches von wissenschaftlichen Werken scheinen bei allen erwähnten Auffassungen der deutschen und schweizerischen Lehre keine nennenswerten Unterschiede vorzuliegen. Abweichungen bestehen nur im begrifflichen Instrumentarium, ferner in der Bedeutung des freizuhaltenden individuellen Bereiches, der als Bestandteil der schutzbegründenden geistigen Gesamtleistung geeignet ist, die Schutzqualität eines konkreten Werkes mitzubegründen. Für den Rechtsuchenden kann die Lösung, die grundsätzlich alle inhaltlichen Elemente der Leistung einbezieht, in gewissen Fällen praktische Vorteile bringen. In allen Fällen kann für die Beurteilung von Verletzungshandlungen nicht darauf verzichtet werden, einen schützbaeren und einen freien Bereich zu unterscheiden. Unterschiede im Ergebnis dürften nur in jenen seltenen Grenzfällen auftreten, wenn ein gemeinfreier individueller Stoff mit einer unpersönlichen Form

90 Im Ergebnis gilt für alle, die der Antithese Form – Inhalt die Antithese individuelle Züge – Gemeingut gegenüberstellen, das Gleiche: Gewisse Ergebnisse menschlichen Schaffens müssen im Interesse des geistigen Lebens auch dann allen Menschen zur ungehinderten Benützung offenstehen, wenn sie aufgrund der Individualität einem Menschen zuzuordnen wären. Im Prinzip folgen der Gegenüberstellung individuelle Züge – Gemeingut im schweizerischen Schrifttum KUMMER M., S. 105ff. und S. 130; RAPP K., Der Anwendungsbereich der Rechte zur Verbreitung und zur körperlichen Wiedergabe im schweizerischen Urheberrecht, S. 18; LUTZ M. J., Schranken des Urheberrechts nach schweizerischem Recht, S. 13.

91 BGE 88 IV 127.

i.w.S. bzw. mit einem unoriginellen inhaltlichen Gewebe vereinigt ist. Das kann der Fall sein bei der Darstellung von Erkenntnissen in Formeln, Tabellen, Zahlenbegriffen oder stereotypen Wortformeln. Nach ULMERS Lehre wäre dann wohl der einfache Wiedergabeschutz für die konkrete Vorlage, jedoch kein Schutz für die abstrakte Erkenntnis oder die Wiedergabe in einem neuen, nachgeschaffenen Mitteilungsträger zu gewähren. Nach der schweizerischen Auffassung würde überhaupt kein Schutz bestehen. Von erheblicher praktischer Bedeutung ist dieser Unterschied schon wegen der Durchsetzungsprobleme nicht.

6. Kapitel: Die urheberrechtliche Beurteilung der einzelnen Softwarebestandteile

A. Softwareergebnisse mit vorwiegend umgangssprachlichen Elementen ¹

I. *Die Form*

Die sprachlichen Mittel werden im allgemeinen in einer stark inhaltlich und fachlich geprägten, eher farblosen Form verwendet. Die Übersichtlichkeit und die Erfordernisse einer schnellen und klaren Verständigung gebieten, dass die Sprache einfach, arm an stilistischen Varianten und formelhaft kurz ist. Die persönliche Ausdrucksweise muss in den Hintergrund treten, der Autor lässt die Sache für sich sprechen. Eine hochentwickelte fachspezifische Symbolik (Zeichen, Symbole, Abkürzungen usw.) reduziert den Spielraum für eine individuelle Darstellungsweise ebenfalls stark.

Wesentliche Unterschiede gegenüber den geschützten Darstellungen aus zahlreichen anderen wissenschaftlichen Gebieten existieren aber nicht. Die hier verwendete Sprache ist wesentlich reicher an Variationsmöglichkeiten als es vergleichsweise in der Mathematik oder Chemie der Fall ist. Der Inhalt des Darzustellenden verpflichtet nicht derart auf eine bestimmte Darstellungsweise, dass es in der Mitteilungsförm nicht zu einer individuellen Auffächerung kommen kann. Freiheiten in der Wortwahl, Satz- und Gedankenfolge, räumlichen Anordnung usw. sind regelmässig vorhanden. Aber auch für die innere Ordnung der Mitteilung sind freie, inhaltlich nicht vorgezeichnete Gliederungsmöglichkeiten in genügendem Mass vorhanden. Nicht von Bedeutung ist, dass die einmal gewählten formalen Elemente später «gebunden» sind, weil Softwarebestandteile der nachfolgenden Entwicklungsphasen bestimmte Wörter, Gliederungen und inhaltliche Strukturen übernehmen und in ein festes, nicht mehr ohne weiteres änderbares Dokumentations- und Programmsystem eingliedern. Entscheidend ist lediglich, dass die erstmalige Auswahl so oder anders hätte getroffen werden können, und nicht, ob die einmal gewählte Form in den nachfolgenden Softwarebestandteilen ohne grosse Mühe wieder geändert werden kann. Die Form wird nicht dadurch zum sachlich richtigen und inhaltlich bedingten Prinzip, dass sie in den späteren Bearbeitungsphasen (Programmierungsplanung, Programmierung) als ein unverrückbares Element beibehalten werden muss, um

1 Siehe Aufzählung oben Kap. 2 E I.

die systematische und terminologische Einheitlichkeit innerhalb eines ganzen EDV-Projektes zu gewährleisten.

Insbesondere bestehen individuelle Möglichkeiten bei der äusseren und inneren Anordnung der Beschreibung von Datensätzen und Datenfeldern, Datensatz- und Datenfeldeigenschaften, Umformungsregeln, Handlungsregeln für den Einsatz des Systems, Fehler- und Verarbeitungsbedingungen (Meldungen), Verzeichnissen von Fachbegriffen, Abkürzungen und Symbolen usw.

In der schweizerischen Lehre dürfte denn auch ausnahmslos für diese problemlose Gruppe von Softwarebestandteilen die Schutzfähigkeit bejaht werden. KUMMER bekennt sich ausdrücklich zum Schutz des umgangssprachlich formulierten Programmes². TROLLER mag entsprechend seiner Auffassung vom Schutz wissenschaftlicher Sprachwerke gleicher Auffassung sein³.

II. Der Inhalt

Der Inhalt dieser Softwarebestandteile ist in umgestalteter Form in den Ergebnissen späterer Entwicklungsstufen, insbesondere auch in den Programmen vollständig enthalten.

Auf die Individualitäten im Bereich des Inhaltes (Konzept, Analyse, Softwarelösung im allgemeinen) wird später⁴ näher eingegangen.

B. Softwareergebnisse mit grafischen oder schematischen Darstellungen⁵

I. Flussdiagramme verschiedener Ebenen⁶

1. Die Form

Die verschiedenen Flussdiagramme unterscheiden sich in ihrer Form nur unwesentlich, weshalb sie hier gemeinsam behandelt werden. Für die grafischen Darstellungen gelten die gleichen Grundsätze wie für technische Zeichnungen und Formulare⁷.

2 KUMMER M., Werk, S. 203.

3 TROLLER A., IG I, S. 421; HEIM E., stat. Einmaligkeit, S. 83, Fn 236.

4 Unten Kap. 6 C.

5 Oben Kap. 2 E II, Beispiele Anhang Nr. 3 und 4.

6 Oben Kap. 2 E II 1.

7 Vgl. KUMMER M., Werk, S. 122ff; TROLLER A., SJZ 60 (1964), S. 369ff.

a) *Zeichnerische, darstellerische Elemente*

Entgegen zahlreichen Auffassungen⁸ besteht bereits in der Darstellungsform Raum für eine individuelle Gestaltung. Ausser Betracht fällt die Form der genormten Symbole und deren rein logisch bestimmte Darstellungsfolge. Alles andere bleibt frei. Der Zeichner ist an keinen Massstab gebunden. Bei System- und Datenflussdiagrammen bleibt insbesondere der freien Gestaltung vorbehalten: Länge, Art und Verknüpfung der Linien, Auswahl von alternativen Symbolen, Verhältnis und Anordnung der einem Hauptsymbol (Aktivität, Programm) zugehörigen Neben-Symbole (Dateien, Belege, Input und Output), Flussrichtungen von und zu den Haupt- und Neben-Symbolen, Abstände zwischen den zeichnerischen Einheiten und den Einzelsymbolen, Trennlinien und weitere formulartechnische Massnahmen zur Verbesserung der Übersicht, Unterscheidung und Führungen von Haupt- und Nebenzweigen im Ablauf, Plazierung und Auswahl von erläuternden Stichworten, Merkmalen und ausführlicheren Kommentaren. Bei Programmablaufdiagrammen kommen hinzu: Wahl der Ein- und Ausgänge bei logischen Entscheidungen, Festlegung der linear verlaufenden Achse und von parallelverlaufenden Nebenachsen, Variationen in der Ausführlichkeit usw.

Bereits im zeichnerischen Bereich sind so zahlreiche Variationsmöglichkeiten vorhanden, dass es in den überwiegenden Fällen für die Erreichung einer handgreiflichen Individualität genügt. Zu beachten ist, dass diese Darstellungen bei weitem nicht gleich straffen Regeln folgen wie z.B. Pläne der Architektur oder des Maschinenbaus. Das Diagramm ist keine massstabgetreue Vorlage für die Herstellung eines körperlichen Gegenstandes.

b) *Sprachliche Elemente*

Ohne Beschriftungen wären die Flussdiagramme nicht zu verstehen, sie bedürfen meistens einiger Erklärungen.

In den meisten Darstellungen sind so viele Wortgefüge vereinigt, dass die Wortkombinationen individuell sind⁹. Die Wortzahl ist so gross, dass die Eigenart der einzelnen Wortkombinationen nicht sehr ausgeprägt sein muss. Eine Vielzahl von Analytikern und Programmierern dürfte sich bereits bei kleineren Diagrammen deutlich unterscheidbarer Wortkombinationen bedienen. Entscheiden muss aber auch hier der Einzelfall.

⁸ Vgl. u.a. KÖHLER R., S. 61; BRUTSCHKE P. G., S. 107.
⁹ Eingehend KUMMER M., Werk, S. 82 (wenige Wörter).

2. *Der Inhalt*

Die schöpferische Leistung kann sich auch im Inhalt ausweisen. Können EDV-Fachleute freie Entscheidungen fällen, sodass für eine Problemstellung unterschiedliche Flussdiagramme entstehen, dann rechtfertigt dies – nach einem Teil der Lehre – ungeachtet der Individualität des Ausdrucksmittels einen Urheberrechtsschutz.

Die Inhalte der verschiedenen Flussdiagramme unterscheiden sich. Das Programmablaufdiagramm entspricht bei einigen Abweichungen in der Detaillierung weitgehend dem Programm. Für dieses sei auf unten Kap. 6 C II verwiesen. Anders verhält es sich mit den anderen Flussdiagrammen. Sie sind neben den oben¹⁰ genannten umgangssprachlichen Beschreibungen des Lösungsweges zusätzliche Hilfsmittel grafischer Art, um den bereits gefundenen oder noch zu findenden Lösungsweg darzustellen.

Darin wird der Lösungsweg nach verschiedenen Gesichtspunkten in einer im Vergleich zum Programmablaufplan wenig detaillierten Form dargestellt. Sie dienen dazu, den Gegenstand der Programmierung einzukreisen und überblickbar zu machen. Grundlagen bilden die in früheren Bearbeitungsphasen ausgearbeiteten Lösungsstudien. Für die Beurteilung des Schutzes von Flussdiagrammen ist nicht zu unterscheiden zwischen dem bereits in den früheren Phasen erarbeiteten Gedankengut, soweit es in das Flussdiagramm übernommen worden ist, und dem, was bei der Herstellung des Flussdiagrammes an schöpferischen Ideen hinzugekommen ist. Zur Individualität der Flussdiagramme gehören sowohl die von früheren Urhebern durch «Originalwerke» vermittelten schöpferischen Gedanken als auch die vom Gestalter des Flussdiagrammes hinzugefügten individuellen Elemente¹¹. Beinahe alle inhaltlichen Elemente der Flussdiagramme fließen wiederum in das Programm ein, falls sie nicht bei einfacheren Aufgabenstellungen oder bei erfahrenen EDV-Fachleuten dort überhaupt zum ersten Mal konkretisiert werden. Eine generelle Aufteilung der im Inhalt der einzelnen Flussdiagramme (den Programmablaufplan dieses Mal eingeschlossen) wahrgenommenen individuellen Bearbeitungsspielräume ist, wie bereits oben¹² erläutert, weder möglich noch erforderlich.

Auch bei den Flussdiagrammen kann dahingestellt bleiben, ob sich das Verbotungsrecht aus der Abhängigkeit von vorbestehenden «Originalwerken» ergibt oder aufgrund der schöpferischen Eigenarten, die erst auf

10 Kap. 2 E I, Ziff. 1, 2, 4.

11 Zutreffend BRUTSCHKE P. G., S. 107.

12 Kap. 2 C II 2 a, b und 5 B II 3 bb.

der Stufe dieses Werkes zweiter Hand hinzugekommen sind^{13 14}. Demnach kann wiederum verwiesen werden auf die Behandlung der in den Programmen vereinigten inhaltlichen Elemente einer ganzen Softwarelösung¹⁵.

II. *Layouts von Ein- und Ausgabemedien, insbesondere Druck- und Bildschirmbilder*¹⁶

1. *Abgrenzung zwischen Ein- oder Ausgabedaten und Elementen der Software*

Daten sind Gegenstand der Datenverarbeitung und gehören grundsätzlich nicht zur Software¹⁷. Auch wenn in besonderen Fällen gewisse Daten schutzfähig wären, berührt das den Schutz der Software nicht. Die Software will unabhängig davon geschützt sein, ob sich durch sie mit bestimmten individuellen oder meist nicht individuellen Datenkonstellationen schutzfähige Erscheinungen (Outputs) erzeugen lassen. Nur die geistig erarbeiteten Materialien für die Organisation der Datenumformungsprozesse und nicht der Gegenstand oder das datenabhängige Ergebnis solcher Vorgänge interessiert im Zusammenhang mit Software¹⁸.

Das will aber nicht heissen, dass alles, was mit Daten, Programmen und dem Einsatz des Computers erzeugt wird, keine Software sei. Soweit damit wiederkehrend gleiche Ein- und Ausgaben erzeugt werden, die als Beiwert oder Raster die zu verarbeitenden Daten begleiten, kommentieren und organisieren, handelt es sich um Software. Diese Ein- und Ausgaben entstehen unabhängig von der Beschaffenheit des Datenmaterials, noch präziser: diese Ein- und Ausgaben sind immer identisch, wenn das Datenmaterial für

13 Im allgemeinen kann davon ausgegangen werden, dass zwischen den Benutzern verschiedener Werkstufen Genehmigungen vorhanden sind.

14 Missverständlich deshalb BGH in GRUR 1956, S. 284 (Rheinmetall-Borsig II): «...um eine (technische) Zeichnung handelt, durch die der Konstrukteur einem schöpferischen Konstruktionsgedanken *erstmal*s (Hervorh. vom Verfasser) darstellerische Form verliehen hat.» Unerheblich ist, ob diese schöpferischen Gedanken an sich oder mit Bezug auf eine bestimmte Veranstaltung zur Wahrnehmbarmachung *neu* sind. Entscheidend ist allein, ob die in der konkreten Form enthaltenen Gedanken individuell sind. Wenn ja, dann ist aufgrund dieser Individualität das Verbotungsrecht auch für jede weitere Bearbeitung oder Umgestaltung einzuräumen. Das gilt allerdings nur so weit, als die nachschaffende Tätigkeit nicht bloss die zur freien Benützung stehenden inhaltlichen Elemente erfasst.

15 Unten Kap. 6 C II.

16 Oben Kap. 2 E II 2 und Beispiele Anhang Nr. 5 und 6.

17 Oben Kap. 2 B II.

18 Dies zur Abgrenzung gegenüber Versuchen KUMMERS (KUMMER M., Werk, S. 205f), aus der Kombination von Ein- und Ausgabedaten Schutzwirkungen für Programme zu begründen. Es handelt sich um Randerscheinungen, die nur ausnahmsweise zu einem Schutz verhelfen können.

ein Programm dieselbe logische Struktur aufweist. Die Ausgabe einer Bilanz auf einem Bildschirm oder Drucker ist unabhängig von konkreten buchhalterischen Daten immer in ein Beiwerk von gleichbleibenden Überschriften, Begleittexten, Trennlinien, Buchstaben- und Zahlengruppierungen (Masken), Zeilen- und Kolonnenanordnungen sowie anderen Hilfsmitteln zur Präsentation des Datenmaterials eingekleidet. Die Programme ermitteln auch in den Bereichen der technischen, wissenschaftlichen und kommerziellen EDV nicht einfach nur Ausgabedaten, sondern verleihen ihnen in den meisten Fällen überdies eine ansprechende und übersichtliche darstellerische Form (sog. Design, Layout). Dasselbe gilt für Eingabedaten, vor allem wenn sie über den Bildschirm erfasst werden.

Diese Darstellungen und textlichen Elemente werden unabhängig von konkretem Datenmaterial bei der Programmentwicklung bereits vorge-dacht, gestaltet und schliesslich in einer Anweisungsform in das Programm eingebracht. Diese Layouts erscheinen in identischer Weise in drei verschiedenen Gestaltungen: in der ersten Skizze als Anleitung für die Programm-planung und Programmierung, im Programmplan bzw. im Programm in Anweisungsform, auf den Ein- und Ausgabegeräten reproduziert bei Gebrauch des Computers in Verbindung mit Ein- und Ausgabedaten.

Eine weitere einleitende Klarstellung: KUMMER¹⁹ macht eine Unterscheidung, die nicht durchführbar ist und zu unhaltbaren Ergebnissen führt. Er geht in seinen Betrachtungen davon aus, dass die einem Programm ange-koppelten Ein- und Ausgabedaten gelegentlich schutzfähig sein können, Programme aber nie, weil sie Anweisungen sind²⁰. Hier wird übersehen, dass Daten- und Programmelemente bis zu einem gewissen Grad austausch-bar sind und bei einem grossen Ermessen des EDV-Fachmannes in eine der beiden Formen gekleidet werden können. Es ist möglich und in gewissen Fällen gebräuchlich, dass ein ganzes individuelles Sortiment von Wörtern²¹, ein längerer umgangssprachlicher Text oder auch eine grafische Darstellung sowohl als Eingabedaten wie auch innerhalb eines Programmes in Anwei-sungsform für die Ausgabe auf Bildschirmen oder Druckern definiert wer-den können. Umgekehrt können Anweisungen in einer beliebigen Textform als Eingabedaten eingerichtet und an ein Programm zur Ausführung ange-koppelt werden. Der Standort in- oder ausserhalb des Programmes kann we-der für den Anweisungscharakter noch für die Schutzfähigkeit entscheidend sein. Einmalige Erscheinungen finden sich hier wie dort.

19 KUMMER M., Werk, S. 204f, auch S. 199.

20 Oben Kap. 5 B II 3.

21 Nach KUMMER M., Werk, S. 199, das Beispiel einer schützba-ren Datenkonstellation.

2. *Die Form der Layouts*

a) *Bildschirmlayouts*

Nicht schutzfähig ist das darstellerische Beiwerk eines Bildschirm-layouts, soweit hardware- und softwaretechnische sowie aus der Aufgabenstellung sich ergebende sachliche Bedingungen oder Beziehungen eine bestimmte Gliederung oder textliche Ergänzung vorgeben. Die Ausgabe einer Bilanz wird z.B. aus sachlichen Gründen stets eine Soll- und Haben-Aufteilung, eine vertikale Gliederung der Einzelkonten und dergl. haben. Andere Gestaltungsspielräume sind schutzfähig. Freiheiten bestehen insbesondere in der Auswahl der aufzuzeigenden Beziehungen und Ergebnisse, in der Wahl und in der Erscheinungsfolge der sie begleitenden umgangssprachlichen Wortlaute²², in der Festlegung von Ausgabemasken und häufig notwendigen Wortkürzungen, in der Vermischung, Reihenfolge, räumlichen und zeitlichen Anordnung der Texte und der auszugebenden Datenelemente, in der Verwendung von weiteren gestalterischen Effekten wie Trennlinien, Spezialzeichen und zeichnerische Symbole, Lichtintensitäten (hell auf dunkel), Lichtstärken bei verschiedenen Datenelementen, Farbauswahl, Schriftgrösse und -typ, Gross- und/oder Kleinschreibung usw.²³. Diese Gestaltungsmöglichkeiten können innerhalb eines einzigen Bildschirmbildes vereinigt sein und beliebig einzelnen Daten- oder Rasterelementen zugeordnet werden. Pro Bild stehen nach der verbreiteten Matrix 1920 Positionen für die Darstellung zur Verfügung. Die Gestaltungsmöglichkeiten beschränken sich aber nicht nur auf ein Bild, sondern ergeben sich auch aus der Zusammenstellung von Bildfolgen oder Bildausschnitten. Vergleiche in der Praxis ergaben, dass selbst bei gängigen und einfachen Aufgaben in gleichen kommerziellen Anwendungsgebieten neben den inhaltlichen Freiheiten augenfällige Unterschiede in den Darstellungen zu beobachten sind.

22 Mit den moderneren, bereits stark verbreiteten Verarbeitungstechniken, insbesondere mit der Dialogverarbeitung, bei der Benützer und Computer via einen Bildschirm ständig kommunizieren, kommt dem Gebrauch der Umgangssprache eine sehr grosse Bedeutung zu. Noch zu Beginn der siebziger Jahre war die umgangssprachliche Kommunikation wenig verbreitet. Damit der Computer zu einer derartigen Kommunikation befähigt ist, enthält die Software umfangreiche umgangssprachliche Textvorräte. Deren Schutzfähigkeit dürfte im allgemeinen über jeden Zweifel erhaben sein. Es dürfte sich erst um den Anfang einer Entwicklung handeln, die weitere Bereiche der Software erfassen wird.

23 Bei älteren Hardwaresystemen sind einzelne Effekte noch nicht möglich, insbesondere Farbauswahlen, Schriftgrössen und -typen. Die Möglichkeiten werden von Jahr zu Jahr grösser. Die Beispiele im Anhang mussten aus fototechnischen Gründen beschränkt werden.

Dank den erweiterten Gestaltungsmöglichkeiten bei den Ein- und Ausgabemedien haben sich den Softwarefachleuten Spielräume eröffnet, die in sehr vielen Fällen Darstellungen zulassen, welche ohne Zweifel individuell sind. Deren urheberrechtliche Beurteilung ist auch einem Laien in den meisten Fällen zuzutrauen. Etwas einfacher als beim Programm ist deshalb auch die Feststellung und Beweisbarkeit der Verletzungstatbestände²⁴. Jeder Programmgebrauch offenbart nämlich diese Layouts dem ganzen Benutzerkreis. Entscheidend ist aber nicht, dass diese Layouts und die sie erzeugenden programmtechnischen Elemente schutzfähig sind, sondern dass deren Schutz auch eine mittelbare Schutzwirkung auf die zugrundeliegenden formbestimmenden Elemente der Programme hat. Der Schutz verlangt, dass durch weitgehende Modifikationen an den Programmen Abstand zu wahren ist.

b) Drucklayouts

Drucklayouts sind ähnlich zu beurteilen wie Bildschirmlayouts. Im allgemeinen sind bei Druckausgaben die gestalterischen Möglichkeiten etwas geringer. Nicht zur Verfügung stehen meistens Farbauswahlen und verschiedene Schriftgrößen oder -typen. Das Ergebnis ändert sich aber kaum: Druckdarstellungen haben sehr oft eine schützbare Individualität.

c) Datensatz- (Record-) Layouts

Minimal sind die Anordnungsmöglichkeiten bei Ein- und Ausgaben, die sich nicht eines Druckers (inklusive Plotters) oder Bildschirmes bedienen. Die gestalterischen Freiheiten (Datenstrukturierung, Masken der Datenelemente usw.) genügen nur ausnahmsweise für eine schützbare Individualität. Programme ohne Bildschirm- und Druckfunktionen enthalten in der Regel keine schutzfähigen Elemente von Seiten des individuellen darstellerischen Beiwerkes. Teile des darstellerischen Beiwerkes können auch ausserhalb der Programme eingerichtet werden²⁵.

3. Der Inhalt der Layouts

Der Inhalt der Layouts ist auch in den Programmen enthalten. Es kann daher auf die Überlegungen in Kap. 6 C II verwiesen werden.

24 Unten Kap. 6 E V 2.

25 Unten Kap. 6 D.

C. Softwareergebnisse in einer maschinell verwertbaren (kunst-) sprachlichen Form²⁶

Aus Gründen der Übersichtlichkeit wird zunächst nur die Rede sein von Source-Programmen in höheren Vielzwecksprachen oder problemorientierten Sprachen mit breitem Einsatzgebiet (allesamt Symbolsprachen). Dem Code dieser Programme kommt eine besondere Bedeutung zu, weil höhere Norm-, Struktur- oder Generatorsprachen oft auf diese Zwischenform zurückgeführt werden. Es können nur die Eigenarten der gängigsten Spracharten²⁷ berücksichtigt werden.

I. Individualitäten im Bereich der Form oder im «Wortlaut»²⁸ der Programme

Entgegen der einhelligen Auffassung in der Literatur ist festzustellen, dass sich im Wortlaut von Programmen, im computer-verständlichen sprachlichen Ausdrucksmittel handfeste Individualitäten nachweisen lassen. Die Gestaltungsfähigkeit dieser Ausdrucksmittel ist jedoch wesentlich bescheidener als in einer umgangssprachlich formulierten wissenschaftlichen Arbeit. Die entgegengesetzte Auffassung geht davon aus, dass bei der Formulierung in einer maschinenverständlichen Sprache «eine Freiheit in der Wahl des «Wortlautes»»²⁹ nicht existiert.

Die Formulierung könne nicht als zweckfrei gelten, «sie sei vielmehr aufs strengste formalisiert und mathematisiert, wie der Computer es erfordert, der nicht die konstruktive Verschwendung aufweist, eine Vielzahl unterschiedlichster Auslösesymbole mit ein und demselben Arbeitsvorgang zu verknüpfen³⁰.»

Zunächst ist festzuhalten, dass über die Menge der verfügbaren Auslösesymbole nicht unmittelbar die Konstruktion des Computers (die Hardware) entscheidet, sondern die zwischen Programmcode und Computer eingeschobenen Softwarekomponenten, insbesondere Compiler und andere Hilfsmittel. Das oben erläuterte Prinzip³¹ der Befehls- und Programmhierarchie erlaubt es, theoretisch beliebig viele alternative Auslösesymbole mit äquivalenten Arbeitsvorgängen zu verknüpfen. KUMMERS Argumentation ist nicht zwingend, jedoch im wesentlichen zutreffend. Selbst die umfangreich-

26 Oben Kap. 2 E III, insbes. Ziff. 2 und Beilage in Anhang Nr. 7.

27 Oben Kap. 2 B I 2.

28 Terminologie KUMMER M., Werk, S. 203.

29 Stellvertretend für die entgegengesetzte Auffassung: KUMMER M., Werk, S. 203f.

30 KUMMER M., Werk, S. 204.

31 Oben Kap. 2 B I 2.

sten Sprachen verfügen nur über ein beschränktes Repertoire an Vorgängen (Operationen) und eine im Vergleich zur Umgangssprache knappe Auswahl an formalisierten Ausdrucksmöglichkeiten. Befehlsredundanzen werden aber, wenn auch eher ausnahmsweise für die Erhöhung des sprachlichen Komforts, regelmässig eingerichtet³². Soweit bei der Auslösung von Funktionen den Regeln von bestehenden Hardware- und Softwareelementen gefolgt werden muss, besteht in der Tat nur in geringem Masse Raum für die freie Wahl von sprachlichen Mitteln. Das gilt jedoch nur für jenen symbolischen Teil eines Programmes, mit dem die ausführbaren Funktionen festgelegt werden. Übersehen wurde aber, dass in den Programmen eine andere, weitgehend frei zu vereinbarende Symbolik existiert, mit der bezeichnet wird, womit, d.h. mit welchen Operanden die ausgelösten Funktionen zu arbeiten haben. Der Computer hat keine Beziehungen zu Sinn und Zweck der abzuwickelnden Arbeitsvorgänge. Die Symbolik für die zu bearbeitenden Datenelemente ist deshalb unerheblich und kann frei gewählt werden³³. Die bereits vorhandene Terminologie des Anwendungsgebietes ist für die Übernahme ins Programm unhandlich, zu wenig präzise, nicht formalisiert und meistens unvollständig. Der EDV-Fachmann muss deshalb für das Arbeitsgebiet und alle vorkommenden Informationseinheiten ein terminologisches System weitgehend neu vereinbaren. Er muss geeignete Wörter aus dem Anwendungsgebiet auswählen oder neue Wörter kreieren, d.h. einprägsame Merkmale den auszulösenden normierten Befehlssymbolen zuordnen. Diese Zuordnung, nicht aber die Merkmalsbildung, ist Bestandteil der Lösung und weitgehend vorgegeben, also urheberrechtlich irrelevant. Die frei gewählte Operandensymbolik ist hingegen aus urheberrechtlicher Sicht von grosser Bedeutung. Aufgrund der Wortauswahl und -bildung entstehen auf der Ebene des Ausdrucksmittels beträchtliche Spielräume für eigentümliche Wortkombinationen. Beliebige viele ähnliche oder vollständig übereinstimmende Programme³⁴ können bei Verwendung der gleichen Programmier-

32 Unten Kap. 6 C II 5.

33 Ein Befehl heisst z.B.: ADD (addiere) A B GIVING C. A, B und C sind Operanden, die beliebig bezeichnet werden können. Es handelt sich um den frei zu vereinbarenden mnemotechnischen Operandenteil eines Befehles. Diese Vereinbarung ist innerhalb eines Programmes zu beachten, indem für ein damit bezeichnetes Datenelement stets das gleiche Symbol zu verwenden ist. Die Operationssymbolik (ADD, GIVING), d.h. der mnemotechnische Operationsteil und der Befehlsaufbau sind die nicht schützbares Regel bzw. die allenfalls in unwesentlichen Einzelheiten schützbares konkrete sprachliche Form einer Programmiersprache. Die Programmierungsregeln müssen in allen Programmen, die in der gleichen Sprache geschrieben werden, befolgt werden und sind vom konkreten Programm zu unterscheiden.

34 Materiell wird es sich immer um das gleiche Programm handeln, denn die Symbolik wird beim Umwandlungsvorgang (Compiler) neutralisiert, indem die Symbole durch konkrete Speicheradressen ersetzt werden. Deshalb wird in dieser Beziehung die Schutzzfähigkeit des Source-Codes, der von primärem Interesse ist, anders zu beurteilen sein als die des Objekt-Codes.

sprache deutlich unterscheidbar formuliert werden, weil die zahlreichen Operandenmerkmale beliebig austauschbar sind und von jedem Programmierer nach seinem Gutdünken festgelegt werden können.

Im einzelnen sind in der Form eines Programmes folgende individuelle Charakteristiken anzutreffen:

1. *Merkmalsbildung, Operandensymbolik, Schlüssel- oder Kennworte*

Innerhalb eines Programmes bzw. eines Bündels von Programmen ist die gesamte Operandensymbolik (Vokabular für die Bezeichnung der Informationselemente) frei wählbar. Hierzu gehören die Symbole für Datenfelder, Datensätze, Dateien, Arbeitsfelder, Konstanten, Verzweigungsadressen und Unterprogramme³⁵. Der EDV-Fachmann wird bei der Bildung der Symbolik immer Rückgriff nehmen auf die vorhandene Terminologie des zu bearbeitenden Arbeitsgebietes. Bereits genannte Gründe zwingen aber dazu, die vorhandene Terminologie zu formalisieren, d.h. vor allem zu kürzen. Viele Merkmale müssen zudem der Phantasie entspringen, weil die vorhandene Terminologie nie so weit geht, dass sie den Erfordernissen von Programmen genügen könnte (Verzweigungsmerkmale, Unterprogramm-Merkmale u. dergl.). Die Merkmalsbildung und -anpassung richtet sich nach verschiedenen Kriterien wie Verständlichkeit, Kürze³⁶, Übersichtlichkeit, Einprägsamkeit usw. Bereits in sehr kleinen Programmen sind mehr als hundert Merkmalsbildungen notwendig. Aufgrund der Auswahl, Anordnungen und Kombinationen dieser Merkmale wird bereits im sprachlichen Ausdruck mühelos Individualität erreicht. Es ist sprachlicher Ausdruck. Eine für Fachleute auf Konvention beruhende, aus sich heraus verständliche Mitteilungsform für einen gedanklichen Gehalt ist gegeben³⁷. Für denselben Arbeitsvorgang sind beliebig viele Wortlaute möglich und wahrscheinlich.

2. *Die Kombination und Reihenfolge der Symbolik*

Die Kombination oder Reihenfolge von Merkmalen ist unschöpferisch, soweit sie durch die sachlich bedingte Reihenfolge der Befehle (im imperativen Programmteil) und innerhalb dieser durch die Reihenfolge entsprechend dem Befehlsformat bestimmt wird. Eine Individualität ist den

35 Als eine besondere Art von Symbolbildung kann die freie Zuordnung von Codebedeutungen angesehen werden.

36 Bei den gebräuchlichsten Compilern sind in der Regel nicht über 30 Buchstaben oder Ziffern pro Merkmal erlaubt.

37 Vgl. CONTU, S. 27: «The instruction that make up a program can be read, understood and followed by a human being».

noch meistens vorhanden, weil verschiedene Kombinationen aus programmtechnischer Sicht indifferent sind und vom Programmierer frei gewählt werden können. Frei zu bestimmen, d.h. nicht durchwegs einem sachlichen Prinzip gehorchen³⁸ im Datendefinitionsbereich (deskriptiver Programmteil) eines Programmes: Die Reihenfolge der Dateidefinitionen, Datensatzdefinitionen und insbesondere die Anordnung der Datenfeldmerkmale innerhalb eines Datensatzes, ferner die Reihenfolge der Definition von Arbeitsfeldern und Konstanten. Aber auch innerhalb des Operationsbereiches (imperativer Programmbereich) eines Programmes sind – wie noch näher zu erläutern sein wird – viele Reihenfolgen indifferent³⁹. Der Wortlaut des Programmes wird auch wegen der eigentümlichen Kombination von freigewählten (Operanden) und sprachlich vorgegebenen Symbolen (Operationen) regelmässig individuell sein.

3. *Programmerläuterungen, Kommentare*

In das (Source-) Programm können und werden regelmässig zahlreiche Bearbeitungshinweise, Erklärungen, Gliederungshinweise oder sonst der Verständlichkeit des Programmes dienliche Dokumentationen in frei wählbaren umgangssprachlichen Formulierungen oder grafischen Darstellungen integriert. Dazu gehören ebenfalls Freiheiten bezüglich der Darstellung des Sourcecodes wie die Wahl von Abständen zwischen Befehlselementen (formatfreie Teile von einzelnen Befehlen) und zwischen verschiedenen Befehlen, Einschiebungen, Seitenaufteilung, Zuordnung von Kennnummern, alternative Reihenfolge und Strukturierung von Befehlselementen usw. Diese Elemente haben keine oder teilweise sehr geringfügige Auswirkungen auf den Programmablauf.

4. *Konstantendefinitionen und sonstige Bestandteile für wiederkehrende Ein- und Ausgaben (Bedienungsaufrufe und Elemente des Beiwerks von Bildschirm- und Druckbildern)*

Das Programm hat neben seiner eigentlichen Funktion, Daten umzuformen, noch Nebenfunktionen, bei denen es mit den Benutzern unter zwei Gesichtspunkten kommuniziert.

Einmal hat es die Ein- und Ausgabe von Daten in einer bestimmten

38 Das kann sich nach oben erläuterten Prinzipien auch auf Elemente einer ersten Festlegung beziehen, die z.B. von einem Analytiker in der ersten Programmierungsphase frei gewählt wurden.

39 Beispiele bei KÖHLER R., S. 64f.

Form zu organisieren⁴⁰. Dafür muss das gestalterische Beiwerk von Bildschirm- und Druckbildern (Texte, Darstellungen) in eine formbestimmende Anweisungsform gebracht werden⁴¹. Auch wenn solche darstellerischen Elemente in Programmen definiert sind (u.a. als normal lesbare Texte), kann ihrem Schutz als Form nach den anerkannten Grundsätzen nichts im Wege stehen.

Weiter kommuniziert das Programm aber auch in Belangen, die seine Handhabung und den Ablauf betreffen. Es handelt sich um kurzgefasste umgangssprachliche Meldungen an Bedienungspersonen und Softwarefachleute. Für derartige Zwecke verfügt das Programm (oder eine ihm zugeordnete Datei⁴²) über umfangreiche umgangssprachliche Textsammlungen, die in ihrer Gesamtheit in Formulierung und Anordnung individuell sind.

5. *Programmstruktur, innere Anordnung*

Unter dem Gesichtspunkt der sogenannten inneren Form verfügt der Programmierer ebenso wie der Wissenschaftler über freie Spielräume für die Anordnung, den Plan der Gedanken. Er kann bei der Gestaltung eines gleichen Arbeitsvorganges mehrere beliebige Abschnitte, Teilabläufe, Verknüpfungen und Verästelungen, d.h. eine frei variierbare Programmstruktur bilden. Das Mittel dazu sind verschiedene Arten von Unterprogrammen (Arbeitsroutinen), die mit Verzweigungsanweisungen verknüpft werden. Diese verknüpfenden Anweisungen haben keinen substantiellen Anteil am Inhalt, sie sind nicht eigentlich Bestandteil des Algorithmus⁴³.

Sie haben lediglich die Übergänge innerhalb der gewählten Struktur und gegebenen Logik sicherzustellen. Sie können je nach der gewählten Struktur so oder anders sein, ohne dass die Softwarelösungen und die in ihr vorhandenen Datenumformungsprozesse sich wesentlich voneinander unterscheiden. Die Entscheidungsspielräume in der Programmstruktur sind so gross, dass sie sich bei der Bearbeitung der Programme (Änderung, Unterhalt) im allgemeinen sehr nachteilig auswirken. Bereits in Programmen mittlerer Grösse (2–4000 Befehle) kann es erhebliche Mühe bereiten, zu lokalisieren, was in welchem Kontext geregelt wird. Es fehlt nicht an Versuchen, diese Freiheiten einzuschränken und dem Programmierer gewisse Strukturen⁴⁴ vorzugeben. Aber selbst bei strenger Normierung der Programmstrukturen gelingt es wegen der Vielfältigkeit der zu behandelnden Aufgaben

40 Ausführlich Kap. 6 B II.

41 Oben Kap. 5 B II 3.

42 Unten Kap. 6 D.

43 Oben Kap. 2 D und Kap. 5 B II 3.

44 Vgl. Stichwort «Programmierungskonvention» bei MÜLLER P. (Hrsg.), 5. Aufl., S. 446; oben Kap. 2 E III 3 und Kap. 6 C I 6.

nicht, alle Lösungselemente (Gedanken) in eine Ordnung zu bringen, die keine Freiheiten mehr zulässt. Mittlere Programme dürften selbst bei eingehender Normierung Individualitäten in der Struktur enthalten.

6. *Programmierungskonventionen, Normierungen, Richtlinien*⁴⁵

Richtlinien, die der Vereinheitlichung von Programmen dienen, unterstehen als Anweisungen keinem urheberrechtlichen Schutz. Dasselbe gilt grundsätzlich auch für die in Ausführung der Richtlinien geschaffenen Programmelemente. Sie sind unschöpferisch. Richtlinien haben vor allem Auswirkungen im formellen Bereich eines Programmes. Deshalb sind sie bereits an dieser Stelle zu besprechen. Anknüpfend an die Überlegungen zur Anweisung im allgemeinen⁴⁶ ist festzuhalten, dass Richtlinien – so umfassend wie in der Fachsprache zur Datenverarbeitung verstanden – nicht gleichzustellen sind mit urheberrechtlich ungeschützten Anweisungen oder Rezepten. Soweit Richtlinien derart beschaffen sind, dass bei ihrer Ausführung noch ein gewisser Spielraum offenbleibt, sind sie ungeschützte Anweisungen. Verdichten sich die Richtlinien aber zu einem Werk, weil ihre Ausführung stets an der in Wort und Strich konkretisierten Form haften bleibt, dann können sie für sich und die Ergebnisse ihrer Ausführungen den urheberrechtlichen Formschutz genießen.

a) *Beispiel zu Ziff. 4 oben*

Richtlinien geben einem Programmierer vor, in welchen Situationen bestimmte umgangssprachliche Bedienungsaufrufe (Systemmeldungen) auszulösen sind. Ist die Auswahl und die sprachliche Form dieser Textsammlung – wie meistens zu erwarten – individuell, dann genießt diese den Formschutz. Übernimmt der Programmierer diese geschützte Textsammlung in sein Programm oder in eine zugehörige Datei, dann besteht der Formschutz auch für die im Programm enthaltenen umgestalteten Texte. Teile des Programmes unterstehen demnach wegen der Abhängigkeit von geschützten Elementen der Richtlinien dem Urheberschutz. Faktisch untersteht dadurch das ganze Programm einem Schutz, weil die eng ineinander verflochtenen geschützten und ungeschützten Elemente nur als Einheit brauchbar sind.

b) *Beispiel zu Ziff. 5 oben*

Richtlinien geben einem Programmierer vor, wie das Programm zu strukturieren ist, d.h. wie die innere Form, die Ordnung der verschiedenen

45 Oben Kap. 2 E III 6.

46 Oben Kap. 5 B II 3.

Lösungselemente beschaffen sein muss. Selbst die strengsten Richtlinien, die für mehr als ein konkretes Programm gelten wollen, lassen noch Entscheidungsfreiheiten zu. Die Richtlinien bleiben ungeschütztes Rezept, eine freie Programmierungstechnik⁴⁷.

Ergebnis

Bei den wichtigsten Programmarten sind entgegen der herrschenden Auffassung bereits im «Wortlaut», in der Darstellung und Ausgestaltung von programmtechnisch praktisch identischen Lösungswegen, in aller Regel zahlreiche Individualitäten festzustellen. Ohne den Rückgriff auf individuelle inhaltliche Elemente ergibt sich, dass Programmen in den meisten Fällen auch ein urheberrechtlicher Schutz zu gewähren ist, wenn man jenen Lehren folgt, die an der Antithese Form – Inhalt festhalten⁴⁸ und nur Individualitäten in der Form berücksichtigen und schützen wollen. Entgegen der herrschenden schweizerischen Lehre, die den urheberrechtlichen Schutz von Computerprogrammen ablehnt⁴⁹, ist festzustellen, dass Programme mit grosser Wahrscheinlichkeit auf einer Ebene individuelle Elemente aufweisen, die zur Form des Mitteilungsträgers gehört. Sie geniessen demzufolge den urheberrechtlichen Schutz. Das Problem der Mitteilungsexklusivität besteht auch bei der Verwendung von gleichen oder ähnlichen Programmiersprachen nicht. Abstand zu wahren ist möglich.

II. Individualitäten im Bereich des Inhaltes von Programmen und von Softwarelösungen im allgemeinen

Vor allem im deutschen Schrifttum wird beim Schutz von Programmen auf individuelle Elemente abgestellt, die sich auf der im wesentlichen nicht geschützten inhaltlichen Ebene befinden. Dabei wurde festgestellt, dass sich die Qualität als einmalige geistige Schöpfung vor allem in der Entwicklung und Logik der Gedankenführung, der Auswahl, Sammlung, Sichtung, Anordnung und Einteilung des Materials ausweisen kann⁵⁰. Hervorzu-

47 Nach KÖHLER R., S. 82, sind Programmierungstechniken ebenfalls frei.

48 Mit KUMMER M., Werk, S. 14, stimmt der Verfasser überein, dass TROLLERS Schichtenlehre grundsätzlich nichts anderes ist als die Weiterführung der Aufteilung in Form und Inhalt; vgl. oben Kap. 5 II.

49 Hervorzuheben ist SIDLER P., S. 32ff, der im Anschluss an die Lehre TROLLERS den Schutz wegen ontologischen Bedenken ablehnt; vgl. dazu ebenso TROLLER A., *Ontologie*, S. 416f. Etwas andere Überlegungen führen schliesslich auch bei KUMMER M., Werk, S. 200ff, zu einer Ablehnung.

50 Für viele andere KOLLE G., GRUR 1974, S. 9 und Zitate Kap. 5 B. Oft dürfte es sich auch bei der Software um Grenzerscheinungen handeln, die als «kleine Münze» des Urheberrechtes durch die Rechtsprechung Schutz finden (vgl. Parallele in der Rechtsprechung des BGer in BGE 88 IV 123, Erw. 1).

heben ist, dass die reine Auswahl eines zu bearbeitenden Stoffes aus vielen vorhandenen urheberrechtlich unerheblich bleiben muss, denn dabei handelt es sich ja nur um die Auswahl der noch zu formenden Materie. Diese Auswahl kann zwar individuell sein, doch äussert sich die Auswahl nicht im ausgewählten Objekt selber. Das Objekt wird durch die Auswahl allein nicht geformt und macht aus sich heraus keine Aussage über die Qualität der Auswahl. Die Auswahl kann hingegen dann urheberrechtlich von Bedeutung sein, wenn das Ergebnis der Verarbeitung von ausgewählten stofflichen Elementen kraft der durch diese Auswahl gewonnenen Kombinationsmöglichkeiten eigenartig ist.

Entsprechend dem Erkenntnisgegenstand⁵¹ der Software sind es folgende Ansatzpunkte, die eine Individualität offenbaren können:

1. *Kenntnisse über das Vorhandensein von Grössen, die mit Hilfe eines Automaten umgeformt werden können*

Diese Kenntnisse zeigen sich in der Auswahl und Zusammenstellung von Grössen und Informationseinheiten in den Ein- und Ausgabeteilen (Datensätze, Datenfelder usw.). Informationseinheiten auf magnetischen Datenträgern, Bildschirm- oder Druckdarstellungen werden aufgrund von Informationsbedürfnissen zusammengestellt, die auf bestehenden betriebswirtschaftlichen Methoden, geschäftlichen, rechtlichen und organisatorischen Erfahrungen oder sonstigen wissenschaftlichen Grundlagen basieren. Soweit es sich um eine mechanische Zusammenstellung von Üblichem und allgemein wissenswerten Angaben handelt, kann dies den Schutz nicht begründen. Eine für die EDV eingerichtete Buchhaltungsanwendung wird stets eine Soll- und Haben-Aufteilung, Buchungstexte, Belegnummern usw. enthalten und ist soweit nicht von besonderer schöpferischer Qualität. Aus folgenden Überlegungen ergibt sich jedoch meistens eine eigentümliche, nicht von selbst gegebene zweckentsprechende Darstellung und Zusammenstellung von Informationselementen: Mit Ausnahme von festen Grundelementen steht einem EDV-Fachmann in den meisten Anwendungsgebieten ein grosses Sammelbecken von begründeten Informationsbedürfnissen für die Auswahl zur Verfügung. Praktische Gründe, insbesondere Realisations- und Betriebskosten, technische Bedingungen oder Kapazitätsüberlegungen verlangen eine Auswahl aus der Unmenge von Möglichem und gleichermassen Wünschbarem. Die Auswahl folgt einer Vielzahl von mehr oder weniger fassbaren Zweckmässigkeitsüberlegungen. Das schliesst jedoch nicht aus, dass die Auswahl auch bei gleichen Entscheidungsgrundlagen einen hohen

51 Oben Kap. 2 D, Bedingungen eines Algorithmus oder Programmes.

Grad an Eigentümlichkeit aufweist⁵². Bei Vergleichen in einfacheren, gebräuchlichen und wissenschaftlich gefestigten Anwendungsgebieten überrascht es, welche Vielfalt an Datensatz-Kombinationen, Datenfeldeigenschaften, Bildschirm-, Druck- und sonstigen Auswertungsanordnungen anzutreffen ist⁵³. Ausgeprägt ist die Vielgestaltigkeit auf dem Gebiet der Standardsoftware (mehrfach verwendbare Software). Hier kann sich der EDV-Fachmann nicht von den Bedürfnissen eines bestimmten betrieblichen Milieus leiten lassen, sondern er muss eine Auswahl aus einer Vielzahl von erwünschten Daten- und Lösungselementen treffen. Daraus ergibt sich ein «Lösungsmix», der als eine von vielen denkbaren Möglichkeiten einmalig sein kann. Diese Kombination, die einen ungewöhnlichen Grad an geschäftlicher Erfahrung und EDV-technischer Kenntnis offenbaren kann, ist geeignet, den Schutz von Softwarebestandteilen zu begründen⁵⁴.

2. *Kenntnisse über das Vorhandensein und die Eigenschaften von Grössen, die mit Hilfe eines Automaten aus bestimmten Ausgangsgrössen gewonnen werden können*

Die unter Ziff. 1 aufgeführten Überlegungen treffen ebenfalls für die Kombination der Ausgangsgrössen zu. Die Einmaligkeit kann sich sowohl in der Auswahl des Inputs als auch des Outputs zeigen. Die Auswahl von Eingangsgrössen bestimmt bei genügend umfangreichen und komplizierten Arbeitsgebieten keineswegs die Ausgangsgrössen derart, dass beim Output keine originellen Anordnungen mehr möglich sind. Aus wenigen, allenfalls nur mechanisch zusammengestellten Grundzahlen (Eingangsgrössen) einer Applikation lassen sich so zahlreiche Kennzahlen oder statistische Grössen ermitteln, dass eine Gestaltung des Outputs mit einer ungewöhnlichen Art von gedanklichen Bezügen und Anordnungen möglich ist.

3. *Kenntnisse über das Vorhandensein von Umformungsprozessen (Operationen) für die Umformung von Eingangsgrössen in Ausgangsgrössen*

Ausgangsgrössen werden aus Eingangsgrössen durch Umformungsprozesse erzeugt, die in einer bestimmten Sequenz⁵⁵ ablaufen. Zu-

52 Solche Auswahlen sind auch auf künstlerischem Gebiet gang und gäbe. Der Filmregisseur hat sich bei gegebenem literarischem Stoff an ein bestimmtes Produktionsbudget zu halten, der Maler hat nur ein beschränktes Bildformat und der Komponist nur einen bestimmten Orchesterapparat zur Verfügung.

53 Der Verfasser stützt sich auf eigene Beobachtungen, die ihm während mehreren Jahren auf dem Gebiet der kommerziellen Datenverarbeitung möglich waren.

54 Analog zu ULMER E., Verlagsrecht, S. 102, und der dort zitierten Rechtsprechung.

55 Nachfolgend Ziff. 4.

nächst ist die Rede von den Umformungsregeln, die im generellen Algorithmus, also nicht in der konkreten programmtechnischen Lösung enthalten sind. Diese Kenntnisse resultieren aus der Analyse von Problemstellungen. Sie können aber ohne weiteres unvermittelt durch den Programmierer in eine konkrete programmtechnische Lösung einfließen. Die Kenntnisse über Umformungsprozesse entstammen wiederum verschiedensten Wissensgebieten. Die schöpferische Eigenart kann sich daraus ergeben, dass entweder bestehende, allenfalls sehr einfache und naheliegende algorithmische Elemente zu einer ungewöhnlichen Gesamtlösung kombiniert werden oder aber dass aus vorhandenen allgemeinen Wissenselementen eine für den Computer taugliche algorithmische Lösung erst geschaffen wird. Für die gebräuchlichen Kostenrechnungssysteme finden sich in der betriebswirtschaftlichen Literatur algorithmische Elemente (Berechnungsformeln usw.). Die Benutzung solcher Lösungselemente kann nur dann schutzbegründend sein, wenn diese bei einer Vielzahl von in Betracht kommenden Lösungsmöglichkeiten mit aussergewöhnlicher Gewandtheit und Kreativität in einer EDV-Lösung kombiniert werden. Solche Eigenarten können bei «massgeschneiderten» und insbesondere bei den für die Bedürfnisse eines grösseren Anwenderkreises geschaffenen Softwarebestandteilen sehr oft erkannt werden. Gerade die Standardprogramme vereinigen eine Vielzahl von denkbaren Lösungsoptionen, die auf der Ebene des Applikationsablaufes, des Programmablaufes, des Datensatzes oder des Datenfeldes «gesteuert» werden können. Derartige Sammlungen von an sich einfachen algorithmischen Elementen können insgesamt eine grosse geistige Leistung von besonderem Gepräge darstellen. Aber auch bestehende Wissenselemente ohne mathematische Form können durch die Auswahl von Annäherungen, Vereinfachungen, Annahmen und Abgrenzungen zu eigenartigen algorithmischen Lösungen entwickelt werden⁵⁶.

Ebenso wie der Algorithmus kann die dafür entwickelte programmtechnische Lösung eine Leistung von eigenartigem Gepräge darstellen. Die Qualität dieser Leistung kann sich in der Auswahl der Programmiersprachen und in der Kombination der einzelnen Befehle zeigen⁵⁷.

4. *Kenntnisse über die logische Ordnung und die Reihenfolge des Ablaufes von automatisierten Umformungsprozessen*

Im Gegensatz zu den Ordnungselementen, die oben in Kap. 6 C I 5 unter dem Gesichtspunkt der inneren Form erfasst wurden, geht es hier um

56 Ein solcher Anwendungsfall könnte bei dem in BGE 98 Ib 396 beurteilten Programm vorhanden sein. Dieses Programm enthält ein Verfahren zur Messung subjektiv qualifizierter Begriffe.

57 Näheres nachfolgend Ziff. 5.

Merkmale, die Bestandteil des system- und programmtechnischen Algorithmus⁵⁸ sind und nicht bloss dem formalen Gerüst eines Programmes zugehören.

a) *Der Ablauf von verschiedenen Programmen, der Applikationsablauf*

Eher bescheiden ist die Entscheidungsfreiheit bei der Organisation des Ablaufes von mehreren Programmen⁵⁹, die im Computer-, Daten- oder Applikationsablaufplan vorgezeichnet wird. Klare Zweckmässigkeitsüberlegungen und technische Regeln legen bei bestimmten Hardware- und Softwarevoraussetzungen den Applikationsablauf weitgehend fest. Gewisse Entscheidungsfreiheiten bleiben bei der Wahl der Dateien- oder Datenbankstruktur, bei der Aufteilung und Abgrenzung der einzelnen Programme⁶⁰, bei der Aufgabenverteilung zwischen den Programmen, beim Ablauf selber durch die Einrichtung von Zwischen- oder Hilfsdateien, Sicherungsabläufen u. dergl.⁶¹.

b) *Der Programmablauf*

Bereits deutlich hervorgehoben wurde in der Literatur⁶², dass bei der Organisation des Programmablaufes eine Fülle von Entscheidungsfreiheiten auftreten. Die in einem Programm vereinigte Menge von mehr oder weniger in sich geschlossenen Lösungselementen und die innere Struktur der einzelnen Lösungselemente (z.B. der Aufbau einer Berechnungsformel) bringen mit sich, dass einzelne Befehle, Befehlsgruppen, Programmabschnitte, Unterprogramme nicht «standortgebunden» sind, d.h. sie können ohne wesentliche Beeinflussung des Programminhaltes an verschiedenen Stellen eingeordnet werden. In vielen Fällen ist es dem Ermessen⁶³ des Programmierers überlassen, ob gleich wirkende Befehlssequenzen im einzelnen so oder anders ablaufen. Neben der logischen Begabung, praktischen Erfahrung und Phantasie entscheidet oft nur der Zufall, in welchem Zusammenhang er einzelne Lösungselemente gedanklich assoziiert. Varianten ergeben sich auch aus der Auswahl von linearer und zyklischer Programmie-

58 Zum Algorithmus oben Kap. 6 C II 3.

59 Es versteht sich, dass diese Unterscheidung schwierig und nur nach eingehender Analyse eines Programmes möglich ist.

60 Die meisten EDV-Anwendungen bestehen aus einer Mehrheit von Programmen, die nur zusammen eine gebrauchsfähige Einheit darstellen.

61 Wahl, ob kleinere oder grössere Programmeinheiten. Wahl der Programmstruktur (Segmentierungen usw.).

62 U.a. KÖHLER R., S. 64, S. 78.

63 Näheres zum optimalen Programm nachfolgend Ziff. 5.

rung, insbesondere aber auch aus der Festlegung der Verzweigungsbefehle (Festlegung der logischen Ausgänge). Oft sind keine genaueren Zweckmäßigkeitskriterien fassbar, ob einzelne logische Varianten in linearer Fortsetzung, in Nebenzweigen oder Unterprogrammen codiert werden sollen. Weitere Entscheidungsfreiheiten bestehen bei der Auswahl und dem Standort von Programmkonstanten, bei der Verwendung von Arbeitsfeldern, gesonderten Arbeitsbereichen, Tabellen u. dergl. Logische Varianten ergeben sich dadurch, dass einzelne Steuerungen mit der Grundlogik oder durch programminterne Steuersymbole (Switches) geregelt werden.

Ohne Berücksichtigung der eigentlichen Umformungsprozesse, den programmtechnischen Algorithmusalternativen, ergeben sich aus der Summe der Entscheidungen, die den Ablauf oder die Ordnung der Umformungsprozesse regeln, Eigenheiten, die auch bei einer identischen Problemstellung und mit strengen Programmierungsrichtlinien auszumachen sind. Erforderlich ist allerdings wieder ein gewisser Programmumfang.

5. *Kenntnisse über Methoden zur Anpassung der erforderlichen Umformungsprozesse an die Eigenschaften eines Automaten*

Diese Betrachtungen befassen sich mit den Alternativen, die bei der Umwandlung von allgemeinen Umformungsregeln (allgemeiner Algorithmus) in eine konkrete programmtechnische Lösung vorhanden sind.

Spielräume ohne zwangsläufige Auswahlen ergeben sich aus der gesteigerten Verwendungsmöglichkeit von einzelnen Befehlen. Einzelne Befehle und oder Befehlsgruppen können austauschbar sein, weil sie gleichwertig sind (Befehls- und Befehlssequenzredundanzen). Ihr Einsatz erzeugt eine identische Wirkung mit Bezug auf die Lösung einer bestimmten Verarbeitungsaufgabe. Durch das Bestreben nach Anwendungsbreite und Flexibilität ist in allen Sprachen ein gewisses Mass an Redundanz bei der Formulierung von Anweisungen wünschenswert und tatsächlich vorhanden. Der Grad der Weitschweifigkeit einer Sprache hängt ab von der Anzahl der Sprachelemente, die verwendet werden können, jedoch nicht verwendet werden müssen, um einen Sachverhalt entsprechend den syntaktischen Regeln eindeutig auszudrücken. Die Kommunikationsfreundlichkeit ist neben der Einfachheit, Erlernbarkeit, Anwendungsbreite ein hervorragendes Beurteilungskriterium für die bessere Mensch-Maschine-Kommunikation. Sie nimmt im allgemeinen durch eine Verkleinerung der aus urheberrechtlicher Sicht wünschbaren Redundanz ab. Alternative Anweisungsformen sind vorhanden bei Programmiersprachen mit elementarem oder ausgedehntem, universellem Befehlsrepertoire und – allenfalls weniger häufig – bei problem-

orientierten Sprachen mit einem umfangreicheren Wortschatz, der direkter auf wiederkehrende Aufgaben eines bestimmten Anwendungsgebietes zielt. Bei problemorientierten Sprachen ist der einzelne Befehl durch den Gegenstand der Programmierung etwas stärker bedingt. Allgemein ist zu beobachten, dass «seit etwa 1970 vor allem wegen der Implementierungsfreundlichkeit und der Erlernbarkeit wieder mehr Wert auf die Einfachheit und Überschaubarkeit»⁶⁴ von Programmiersprachen gelegt wird, die für die Erreichung einer besseren Eingabeflexibilität im allgemeinen in eine formatfreiere Form gekleidet werden, d.h. dass z.B. Stellungs- und Operandenmerkmale nach Bedarf in einer freieren Anordnung wahlweise verwendet werden können. Das Befehlsrepertoire bleibt eher klein, der einzelne Befehl geht auf in eine vom Programmierer für den Einzelfall gewählte Anordnung von mehreren Befehlen.

Auf die Verschiedenartigkeit der von einzelnen Programmierern gewählten programmtechnischen Lösungen weist auch hin, dass selbst bei genauer Programmvorgabe Programme verschiedenen Umfanges (Anzahl Programmstatements) mit verschiedenen Laufzeiten entstehen, wobei Programmumfang und Laufzeit nicht notwendigerweise in einer bestimmten Richtung korrelieren.

Auf der Grundlage des gleichen Algorithmus entstehen Programme unterschiedlicher Qualität⁶⁵. Qualitätsmerkmale sind u.a. Ablaufgeschwindigkeit, Kapazitätsbelastung bei den verschiedenen Systemkomponenten, Einfachheit und Komfort in der Bedienung, Herstellungsaufwand, Widerstandsfähigkeit in kritischen Situationen, Übersichtlichkeit und Verständlichkeit. Die meisten Merkmale beeinflussen alle übrigen und stehen oft in einem Gegensatz zueinander. Man kann nicht das Maximum nach allen Richtungen gleichzeitig erreichen. Angestrebt werden kann nur ein Optimum bei einer bestimmten Prioritätsordnung. Soweit Zweckmässigkeitsüberlegungen gewisse Prioritäten im allgemeinen und bei einzelnen konkreten programmtechnischen Massnahmen klar befehlen, besteht kein schöpferischer Spielraum. Es kann davon ausgegangen werden, dass jeder EDV-Fachmann – wenn auch nicht explicit definiert – bestimmten Prioritäten folgen muss. Auch eine klare Prioritätenordnung kann indessen nicht mehr als einen Rahmen geben und keinesfalls einen durchgehend gültigen Leitfaden, der zwangsläufig die zahlreichen system- und programmtechnischen Massnahmen bis auf eine oder wenige Varianten reduzieren könnte. In

64 Schweiz. Hochschulzeitung Nr. 83 vom 5.5.79, ein Beitrag zu der Tätigkeit von WIRTH N., (ETHZ), Zürich. Eingehendere Überlegungen WEICHELBAUMER C., Methoden und Hilfsmittel zur Entwicklung von Benutzersprachen, Data Report (Siemens), H. 3/79, 1979, S. 40ff, S. 45.

65 ZELLE/SCHLECHTNER/SCHMID, S. 9.

überwiegendem Masse bleiben schöpferische Freiheiten erhalten, weil innerhalb eines Programmes eine bestehende generelle Prioritätsordnung bewusst oder unbewusst mehrmals situationsgerecht variiert werden kann oder muss. Die Befolgung einer Ordnung muss schon oft daran scheitern, dass exakte Entscheidungsgrundlagen nur mit unverhältnismässigem Aufwand erhoben werden können oder nicht quantifizierbar sind. Selbst wenn man sich bei statistisch erfassbaren Grössen (Beschaffenheit des Datenmaterials, Ausführungszeiten von Befehlen usw.) nur mit Annäherungswerten begnügen wollte, blieben noch mehrere unquantifizierbare Faktoren (Einfachheit und Komfort bei der Bedienung, Übersichtlichkeit usw.), die noch genügend freie Entscheidungen dem Programmierer überlassen. Schliesslich blieben auch in der theoretischen Situation, dass für alle Entscheidungen die Prioritäten und Beurteilungsfaktoren einigermassen quantifiziert werden könnten, äquivalente Alternativen offen (z.B. Vertauschung von gleichen Befehlen). Die Tatsache, dass die Aufgabenstellung des Software-Fachmannes grundsätzlich vom Gedanken der Optimierung getragen ist, darf nicht zur Annahme verleiten, dass die zahlreichen Wahlmöglichkeiten im Rahmen eines umfangreicheren Programmes durch einen entsprechenden geistigen Einsatz auf wenige gangbare Lösungswege reduziert werden könnten.

Unrichtig ist deshalb, wenn auch ausserhalb von ganz trivialen Programmen in Betracht gezogen wird⁶⁶, dass eine unbegrenzte Zahl von Programmierern bei der Lösung einer bestimmten Aufgabe zu identischen Programmen gelangen würde, wenn man ihnen nur unbegrenzt Zeit liesse. Das sogenannte optimale Programm ist nur von theoretischem Interesse⁶⁷, und die Annahme ist hypothetisch, dass auch im Bereich von einigermassen optimalen Programmen nur eine kleine Zahl von gleichwertigen Alternativen zur Verfügung stehen. Derartige Anforderungen werden in der Praxis auch selten gestellt, weil die Entwicklung von stark optimierten Programmen zu aufwendig und im Verhältnis zu anderen Kostenfaktoren uninteressant ist. Bedenken wegen der Monopolisierung von optimalen Programmen sind nicht am Platz.

Zusammenfassung

Zutreffend ist, wie bereits im Jahre 1967 von ULMER⁶⁸ festgestellt, dass bei Programmen (und bei Softwarebestandteilen im allgemeinen) in inhaltlicher Hinsicht eine Breite von Wahlmöglichkeiten besteht, die notwen-

66 SIDLER P., S. 32ff.

67 Dazu näheres bei KÖHLER R., S. 65–67, S. 82f. KÖHLERS Ausführungen sind im Ergebnis zutreffend.

68 ULMER E., Wiss. Werke, S. 17.

digerweise dazu führt, dass je umfangreicher das Programm (oder das ganze EDV-Projekt) wird, um so mehr die von verschiedenen Programmierern (oder EDV-Fachleuten) für dieselbe Aufgabe entwickelten Ergebnisse voneinander abweichen.

III. *Der Schutz von Programmen in verschiedenen Sprachen*

Die vorangegangenen Betrachtungen beziehen sich auf die für Anwenderprogramme gebräuchlichsten höheren Symbolsprachen⁶⁹. Anderen Sprachen kommt gemäss den neuesten Erhebungen⁷⁰ immer weniger Bedeutung zu. Ihre Charakteristiken lassen sich auch kaum allgemein zusammenfassen. Eingehendere Betrachtungen erübrigen sich also im folgenden.

I. *Maschinenorientierte oder maschinennahe Sprachen, einfache Symbolsprachen, Assemblersprachen u. dergl.*

Aus urheberrechtlicher Sicht fallen zwei Unterschiede ins Gewicht. Anstelle von frei vereinbarten Operandensymbolen müssen oft direkt Speicheradressen (Zahlenwerte) eingesetzt werden. Die Symbole, die Kommentierungsmöglichkeiten und sonstigen formalen Merkmale, insbesondere die Eingabeflexibilität und die Formationen innerhalb des Sourcecodes sind starrer oder beschränkter. Individualitäten in der Form, im «Wortlaut» des Programmes, sind gelegentlich nicht mehr vorhanden. Nicht unbedingt zutreffend und von Bedeutung ist, dass diese einfacheren Sprachen mehr Individualitäten zulassen bei der Kodierung, weil deren Anwendung nicht wie bei den problemorientierten Sprachen so stark durch den Gegenstand der Programmierung bedingt ist⁷¹ und deswegen die Variationsmöglichkeiten für gleichwertige Befehle oder Befehlskombinationen grösser sind. Umständlicheres Programmieren will nicht unbedingt individuelleres Programmieren sein. Auch mit solchen Programmiersprachen werden Probleme eines bestimmten Anwendungsgebietes gelöst, und ein geübter Programmierer wird sich zuerst dem Problem entsprechende, mehrfach verwendbare Befehlssequenzen (sog. Makros) zurechtlegen, die einem oder mehreren Befehlen einer höheren problemorientierten Sprache entsprechen. Solche Befehlssequenzen sind in der Regel stereotype handwerkliche Mittel und gehören deshalb nicht zum individuellen Bereich.

69 Oben Kap. 2 B I 2 und Kap. 6 C.

70 Oben Kap. 2 B I 2; ZIMMERMANN W. L., Bd. 2, S. 286.

71 KÖHLER R., S. 87; KOLLE G., GRUR 1974, S. 9 und dort zit. Literatur.

2. *Struktur- und Generatorsprachsysteme*⁷²

Struktur- und Generatorsprachsysteme erfreuen sich zunehmender Beliebtheit für verschiedene Aufgaben. Diese Systeme weisen die Vorteile grösstmöglicher Problemnähe auf. Die Spielräume in der Kodierung, der programmtechnischen Realisierung sind geringer, weil der Generator nach bestimmten formalen Regeln mehr oder weniger direkt in umfassenden Anweisungen⁷³ das allgemeine Lösungsprinzip übernimmt. Typische Anwendungsfälle für Generatorsysteme sind Aufbau und Unterhalt von Dateien und Datenbanken, Aufbau und Vorbereitung von Datensätzen für die Ein- und Ausgabe in Bildschirmbildern, Ausgabe von Druckbildern, Sortierungs- und Mischvorgänge, Aufbau von Kommunikationsnetzen usw. Nur in Ausnahmefällen besteht eine EDV-Anwendung ausschliesslich aus Anweisungen auf der Ebene von Generatorsprachen. Immer sind noch zusätzliche Programme in den gängigen Symbolsprachen⁷⁴ erforderlich. Selbst wenn der Schutz der Generatorsprachprogramme in einzelnen Fällen nicht gewährleistet werden kann, berührt das den Schutz einer ganzen gebrauchsfähigen EDV-Anwendung, eines Systems von mehreren Programmen, kaum. Einfachere Generatoranweisungen (z.B. Sort- oder Mischparameter mit blossen Zahlenangaben) haben einen geringen Eigenwert. Bei komplizierteren Generatorsystemen sind verschiedene Ansatzpunkte für einen Schutz vorhanden. Eher bescheiden sind die Möglichkeiten im Bereich der Ausdrucksmittel. Allgemeine Aussagen sind bei so verschiedenartigen Systemen jedoch nicht möglich.

Weit wichtiger ist, dass stets individuelle formale und inhaltliche Elemente der allgemeinen Problemlösung direkt in das Generatorsprachprogramm eingehen und als umgestaltete Ergebnisse der Vorstufen fortwirken können. Es sind dies u.a. die in die Generatorsprache übertragenen Definitionen von Bildschirmbildern, Druckbildern, Daten- und Dateistrukturen, Entscheidungstabellen usw.⁷⁵. Nach dem Grundsatz der Insichverständlichkeit hat bei der Beurteilung die bedeutende geistige Leistung bei der Entwicklung des Generatorsystems und des dem Generator zugrundeliegenden Programmes ausser Betracht zu bleiben. Die Generatorprogramme, eingeschlossen die Compilerprogramme für die üblichen höheren Programmier-

72 Oben Kap. 2 E III 3, 4.

73 Diese Anweisungen werden etwa als Makro-Instruktionen, Parameter oder Steuerangaben bezeichnet, je nach dem, ob der Generator nach dem Prinzip eines Interpreters, Generators oder Übersetzers arbeitet.

74 Oben Kap. 2 B I 2.

75 Oben Kap. 6 A und 6 B.

sprachen, sind nicht als System (Verfahren), hingegen als konkretes Programm für sich nach den oben dargelegten Grundsätzen schutzfähig. Ebenfalls schutzfähig ist das aus der Verbindung von Generatoranweisungen und Generatorprogramm gegebenenfalls erzeugte Programm. Der Schutz kann sich dabei entweder aus der Individualität der nach bestimmten Anwendungsbedürfnissen zusammengestellten eigenen oder aus den im Generator bereits enthaltenen fremden Programmteilen ergeben. Mündet die Vereinigung von Steueranweisungen und Generatorprogramm bei Generatoren mit interpretierenden Elementen nicht in ein neues Programm, weil die Steueranweisungen nur den besonderen Ablauf eines bereits bestehenden Programmes einrichten, dann kann der Schutz wiederum entweder bei den Steueranweisungen oder beim gesteuerten Programm einsetzen. In deren schöpferischen Verbindung zu einer Einheit kann allenfalls ein Werk zweiter Hand⁷⁶ gesehen werden. Die Auffassung von KÖHLER⁷⁷, dass die Steueranweisungen (Quellenprogramm) für Generatorsysteme keinen Schutz geniessen, ist nur bei sehr einfachen Systemen, wie sie dieser Autor gemäss den von ihm aufgeführten Beispielen im Auge hat, zutreffend. Mehrheitlich dürften Generatorsysteme individuelle Schaffungsmöglichkeiten zulassen. Man vergleiche z.B. unabhängig voneinander geschaffene Anweisungsgruppen eines Lohnabrechnungssystems⁷⁸, die für die Steuerung eines Generators Berechnungsformeln, Texte und sonstige darstellerische und statistische Elemente enthalten.

IV. *Der Schutz von verschiedenen Programmarten*

1. *Unterprogramme, Programmteile*

Aus urheberrechtlicher Sicht kann nur ins Gewicht fallen, dass die verschiedenartigen Programmteile sich von den selbständigen Programmen in der Regel durch den bescheideneren Umfang unterscheiden. Alle anderen

⁷⁶ VE II Art. 2 URG.

⁷⁷ KÖHLER R., S. 91.

⁷⁸ Hinzuweisen ist auf einen terminologischen Unterschied. KÖHLER R., S. 9 und S. 81, behandelt z.B. eine Lohnsteuertabelle als Referenzinformation, auf die ein Programm bei der Verarbeitung (Run time) zurückgreifen muss. Er betrachtet die Referenzinformationen im allgemeinen aufgrund von Auswahl und Anordnung zu Recht als schutzfähige Werke. Das Kriterium, ob der Rückgriff auf Steueranweisungen vor oder während dem Programmablauf erfolgt, ist nach anderer Terminologie (vgl. MÜLLER P. (Hrsg.), 5. Aufl., S. 221) nicht dafür massgebend, dass ein Programm als Generator (Interpreter) und die Steueranweisung als Teil einer Generatorsprache bezeichnet werden kann. Ob Generatorsprache oder Referenzinformation ist in der Sache auch unwichtig. Beide müssen ähnliche Angaben enthalten, um in gleichem Umfang den Zweck erfüllen zu können, ein bestehendes Programm mit ergänzenden Steueranweisungen für eine konkrete Aufgabe einzurichten.

Eigenschaften sind im wesentlichen gleich. Für den Schutz von Programmteilen gilt in der Regel nichts anderes als für die selbständigen Programme⁷⁹. Probleme können sich allenfalls bei sehr kleinen Programmfragmenten ergeben, weil sie nur unbedeutende Eigenarten aufweisen und kein in sich verständliches und sinnvolles Werk darstellen. Aus praktischen Überlegungen ist das jedoch nicht von Bedeutung. Nur eine Mehrheit von Programmteilen, die in ein einheitliches System eingeordnet ist, kann in der Regel einen Nutzen bringen. Die gegenseitige Abhängigkeit der einzelnen Programmteile ist wegen vieler gemeinsamer Grundlagen wie Datensätze, Datenfelder, Schnittstellenabgrenzungen usw. so stark, dass die Übernahme von einzelnen «freistehenden» Programmteilen uninteressant und harmlos ist. In solche «Lücken» des Urheberschutzes kann der «Selbstschutz» einspringen. Schutzzfähig ist meistens eine Sammlung von Programmteilen in Programmbibliotheken oder eine in einem Programm vereinigte, in sich geschlossene Mehrheit von Programmteilen. Auch eine sehr «lockere» Sammlung von Programmteilen, die einzeln nicht schutzzfähig sind, dürfte geschützt sein, wenn die Art der Zusammenstellung und das der Sammlung zugrundeliegende Konzept von selbständigem Gepräge ist.

2. *Objektprogramme oder maschinell verwertbare Zwischenformen*

Sourceprogramme, die direkt oder mit Unterstützung eines Generators (Source-Generators) in einer der bisher besprochenen Symbolsprachen erstellt werden, müssen vor ihrem Gebrauch in die computerinterne Sprache umgewandelt werden. Sämtliche Anweisungen werden dabei in eine auf dem Dualsystem basierende Ziffersprache (Maschinencode) umgeformt. Es stellt sich die Frage, ob sich gegenüber dem Sourcecode der Bestand an individuellen Elementen im Maschinencode verändert. Das ist zu bejahen:

Es fallen einige Individualitäten aus dem Bereich der Form weg⁸⁰. Der Compiler ersetzt die freie Operandensymbolik durch Zahlen, die auf Speicheradressen hinweisen. Individuell bleibt dabei nur noch die Kombination und die Reihenfolge der durch die Symbolik vertretenen Informationselemente. Programmerläuterungen und Kommentare fallen im Maschinenprogramm vollständig weg. Konstantendefinitionen und sonstige Elemente für wiederkehrende Ausgaben bleiben gleich. Elemente der inneren

79 Die Selbständigkeit der einzelnen Programme ist meistens noch aus dem Grunde beschränkt, weil eine EDV-Anwendung sich sehr oft aus mehreren Programmen zusammensetzt und erst diese Programme als ganzes System eine sinnvolle und gebrauchsfähige Einheit ergeben. Deswegen ist es denn auch nicht erforderlich, dass jedes einzelne Programm und jeder Softwarebestandteil schutzzfähig ist.

80 Oben Kap. 6 C I.

und äusseren Form bleiben, letztere in etwas reduziertem Umfang, demnach weitgehend erhalten. Anerkannt ist, dass bei der Umwandlung inhaltliche Individualitäten⁸¹ des Sourceprogrammes, die zu einem grossen Teil auf die Flussdiagramme und andere Analyseunterlagen zurückgehen, unverändert in das Objektprogramm eingehen⁸².

Neu kommen im Objektprogramm Eigenschaften des Compilers⁸³ hinzu. Im Objektcode werden einzelne Befehle des Sourceprogrammes in gegebenenfalls mehrere Maschinenbefehle aufgelöst und entsprechend verknüpft. Die vom Compiler beigefügten Elemente können individuell sein und ins Objektprogramm eingehen. Wie bei den anderen Generatorprogrammen werden gleichzeitig Rechte von Urhebern des Sourceprogrammes und des Umwandlungs- oder Übersetzungsprogrammes berührt.

Ein Urberschutz für das Objektprogramm ist somit ebenfalls meistens gewährleistet. Die vom Computer erzeugte mechanische Umgestaltung enthält in Form und Inhalt beinahe sämtliche individuellen Elemente früherer Werkstufen. Von geringer praktischer Bedeutung sind die dem Compiler entnommenen Individualitäten, weil der Gebrauch des Compilers im allgemeinen jedermann zugänglich gemacht wird.

Es ist noch klarzustellen, dass die auf die Maschine ausgerichtete Umformung des Programmes in Binärziffern den Schutz nicht in Frage stellen kann⁸⁴. Wie bei Magnettonbändern und Schallplatten handelt es sich um eine Aufzeichnungsform, die mit geeigneten technischen Einrichtungen jederzeit wahrnehmbar und verständlich gemacht werden kann. Auf einem Drucker oder Bildschirm kann das Objektprogramm in binärer, hexadezimaler oder in einer anderen Form sichtbar gemacht werden⁸⁵. Eine Ungleichbehandlung von Source- und Objektprogrammen kommt in vielen Fällen schon deshalb nicht in Betracht, weil bei gewissen Computersystemen und -sprachen die beiden Programmtypen ineinander aufgehen oder sonst nicht mehr deutlich unterscheidbar sind.

81 Oben Kap. 6 C I.

82 Siehe z.B. BRUTSCHKE P. G., S. 109.

83 Siehe analoge Überlegungen bei den Generatorprogrammen oben Kap. 6 C III 2.

84 «Auch ein blosses Zahlenwerk kann dem Erfordernis der sprachlichen Mitteilung genügen» (MÖHRING/NICOLINI, Kommentar zum Urheberrechtsgesetz, § 82 UhrG Anm. 3B mit Hinweisen). Der durch Zahlen ausgedrückte Gedankeninhalt kann aufgrund einer Konvention über den unmittelbaren Bedeutungsgehalt dieser Zeichen hinausgehen und in Form und Inhalt individuell sein.

85 Zusammenfassung der Kontroverse bei BRUTSCHKE P. G., S. 110.

D. Softwarebestandteile mit üblicherweise nur mittelbar wahrnehmbarer Verkörperung auf magnetischen Datenträgern⁸⁶

Die hier einzuordnenden Softwarebestandteile können im Rahmen ihres normalen Gebrauches oder durch besondere Vorkehrungen ganz oder teilweise unmittelbar wahrnehmbar gemacht werden. Unerheblich ist, ob die unmittelbar wahrnehmbare Form die Übliche oder, wie bei diesen Werken, die weniger häufige Form ist. Bei jeder Erscheinungsform gehört das Werk nicht mehr ausschliesslich dem Geiste des Urhebers an und bedarf deshalb eines Schutzes.

I. Referenz-, Parameter-, Textdateien und sonstige Dateien mit programmergänzenden oder -ersetzenden Funktionen⁸⁷

Sämtliche Programmarten, insbesondere aber Generatorsysteme können mit derartigen Dateien gekoppelt sein. Der Rückgriff auf diese Dateien kann vor oder während des Programmablaufes erfolgen. Im Grunde handelt es sich bei diesen Dateien nur um eine Separierung von Elementen, die ihrer Funktion gemäss ebensogut in den Programmen enthalten sein könnten. Nur technische Gründe (Systembelastung, rationellere Änderbarkeit usw.) sprechen für diese Trennung. Es stellt sich die Frage, ob diese Dateien und die ihr zugehörigen Programme als Einheit zu beurteilen sind.

Eine differenzierende Lösung ist am Platze. Die Unterscheidung ist aber wieder nur auf der Werkstufe und nicht auf der Urheberschaftsseite zu machen⁸⁸.

Zu bejahen ist eine einheitliche Würdigung, wenn eine untrennbare Verbindung vorhanden ist und keinem Bestandteil eine Selbständigkeit zukommen kann, weil die Trennung zugleich die Existenz beider Bestandteile zerstören würde und sie nur unbrauchbare Bruchstücke blieben. Das trifft bei Programmen zu, die auf Textdaten und sonstige Angaben zurückgreifen und nicht beliebig auswechselbar sind. Diese Dateien haben für das entsprechende Programm eine unentbehrliche ergänzende Funktion. Typische Anwendungsfälle sind Programme mit Ein- und Ausgaben auf Bildschirmen, Druckausgaben und anderen Verarbeitungsfunktionen, bei denen von Fall zu Fall auf die notwendigen Textvorräte und -konstanten zurückgegriffen werden muss. Programme und Daten bilden eine Einheit. Ihre Schutzfähigkeit kann sich aufgrund der Individualität des Programmteiles und der originellen Auswahl und Anordnung der beigeordneten Datenteile ergeben.

⁸⁶ Oben Kap. 2 E IV.

⁸⁷ Oben Kap. 2 E IV 1 und Kap. 6 C III 2.

⁸⁸ Oben Kap. 2 C II 2 b.

Anders verhält es sich, wenn zwischen den Programmen und den beigeordneten Daten nur eine lockere Beziehung besteht. Diese Daten werden zwar immer genau aufeinander abgestimmt sein müssen. Sie vermögen jedoch nicht für das Ganze einen Schutz zu begründen. Das schliesst allerdings nicht aus, dass nach den besprochenen Kriterien sowohl beim Programm wie auch bei der Datei schutzbegründende Eigenschaften vorhanden sind. Eine solche Beurteilung ist meistens bei Generatorsystemen am Platz.

II. Testdateien

Wie bereits erwähnt⁸⁹, kann auf die Untersuchung dieser wenig gefährdeten Objekte verzichtet werden.

E. Schutzzumfang und Schutzwirkung

I. Einleitung

Wie die vorangegangenen Betrachtungen zeigen, bestehen bei allen Softwarebestandteilen in Form und Inhalt zahlreiche Ansatzpunkte, die den urheberrechtlichen Schutz begründen können. In Übereinstimmung mit der vorherrschenden Auffassung im ausländischen⁹⁰, insbesondere auch im

89 Oben Kap. 2 E IV 2.

90 In den USA besteht die beinahe einhellige Auffassung, dass Computerprogramme in einem wirksamen Masse urheberrechtlich geschützt sind. Ältere und neuere Literatur oder Übersichten bei KÖHLER R., S. 95ff; THURMANN V. R., Der Rechtsschutz von Computerprogrammen in den USA, GRUR Int. 1969, S. 207ff; AXTER H. und O., BB 1967, S. 606, Fn 4 und 5; LAWLOR REED C., The American View, Brighton Conference 69, Report, S. 22ff; Sehr informativ: MOOERS C. N., ACM Computing Surveys, Vol. 7, No. 1 (1975) S. 45ff mit Literatur auf S. 72; Final Report of National Commission on Technological Uses of Copyrighted Works (CONTU), Excerpt mit weiteren Hinweisen in Performing Arts Review, Vol. 9, 1979; Literaturhinweise bei KOLLE G., DdA 1977, S. 78, Fn 35f; ders., GRUR 1973 S. 611ff und GRUR 1974, S. 7f; das US-Copyright Office (reine Registerbehörde) nimmt Computerprogramme für die Registrierung entgegen, vgl. Circular 61 from Copyright Office, GRUR Int. 1964, S. 635; CARY G. D., GRUR Int. 1964, S. 623ff, BANHAF JOHN F., ACM Communications No. 8, 1968, S. 548f. Von dieser Möglichkeit wird in zunehmendem Masse, jedoch noch sehr zurückhaltend Gebrauch gemacht: MYERS E. D., Datamation 3/78, S. 125. Eine besondere Lösung der Probleme im Zusammenhang mit Computern wurde für den neuen Copyright Act ausgearbeitet, jedoch schliesslich nicht realisiert, weil die Zeit noch nicht reif sei und hierzu noch umfangreiche Studien erforderlich seien. Bei Werken im Zusammenhang mit Computern bleibt alles beim Status quo. Rechte, die nach dem bisherigen Copyright Act, nach Common Law oder nach den Gesetzen eines Einzelstaates geschützt waren, bleiben vorbehalten, vgl. v. WESTERHOLT-WEISSTHANNER M., Überblick zum neuen amerikanischen Urheberrecht, in UFITA Bd. 82 (1978), S. 25 und § 117 des neuen U.S. Copyright Act von 1976.

Andere Länder:

England: NEUMEYER F., GRUR Int. 1966, S. 522ff; The Whitford Report, Report of the Committee to Consider the Law on Copyright and Designs, H.M.S.O., 1976.

Canada: Comments on the Copyright Section of the Report of the Economic Council

deutschen Schrifttum⁹¹, ist als Ergebnis zu bestätigen, dass praktisch alle Softwarebestandteile in irgendeiner Form dem Urheberschutz unterstehen. Nicht schutzbegründend sind Softwareelemente, die auf mechanischer Fleissarbeit, blossen Zwangsläufigkeiten und naheliegenden Zweckmässigkeitsüberlegungen beruhen. Ausser Betracht fallen somit alle Konstellationen von Lösungselementen, die von mehreren Personen unabhängig voneinander gefunden oder gleich gestaltet werden können. Ungeschützt sind nach unbestrittener Auffassung die wesentlichen inhaltlichen Bestandteile, insbesondere die allgemeine Lösungsidee und die Algorithmen auf den verschiedenen Lösungsstufen. Ausserhalb des schützbaeren Bereiches liegen alle Elemente, die sich wegen ihres Verfahrenscharakters aus der konkreten Festlegung herauslösen lassen. Frei ist die Benutzung aller Softwarebestandteile für die Analyse der Lösung und die Übernahme der erwähnten wesentlichen inhaltlichen Elemente. Der Schutz ist sehr beschränkt und hält sich, wie sich ein vom Verfasser konsultierter EDV-Fachmann äusserte, an «Nebensächlichkeiten», die aber ebenfalls Ergebnis einer aufwendigen und fehleranfälligen «geistigen Kleinarbeit» sind.

Für einzelne Softwarebestandteile, vor allem bei sehr kleinen Programmen, kann der Schutz vollständig ausbleiben. Meistens ist jedoch im schützbaeren Bereich eine individuelle Färbung vorhanden. In aller Regel ist die blosser Wiedergabe einzelner Softwarebestandteile eine Urheberrechtsverletzung. Auch bei der Bejahung eines minimalen, auf die äussere Form begrenzten Schutzes ist diese untere Verbotsgränze gegeben. Die bedeutendste Verletzungsgefahr, die unmittelbare Verwertung der Leistung durch die Übernahme der konkreten Form, kann durch den Urheberschutz gebannt werden. Der Schutz geht aber, wie in Kap. 6 E II und III zu zeigen sein wird, weiter.

Es ist unrichtig, aus einem sehr begrenzten und lückenhaften Schutzzumfang auf der rechtlichen Ebene eine geringe tatsächliche Schutzwirkung zu folgern. Aus der Natur des schützbaeren Gegenstandes⁹² ergeben sich folgende drei wesentliche Gesichtspunkte, die für eine weitergehende Schutzwirkung sprechen:

of Canada on Intellectual and Industrial Property of the Patent and Trademark Institute of Canada, Toronto 1975.

Schweden: NEUMEYER F., GRUR Int. 1966, S. 417ff.

Diverse: Berichte der verschiedenen Landesgruppen im Rahmen der AIPPI zur Frage 57 B in den Jahrbüchern 1970 folgende: HEIM/CURCHOD, Schw. Mitt. 1972 H. 2, S. 105ff; KOKAI T., Data Exchange 7/8 1975, S. 16ff, Beiträge im Report zur Brighton Conference 1969; DE BRABANTER M. et VANDERPERRE R., L'ingénieur-conseil 1976, S. 137ff.

91 Oben Kap. 1 B und Literaturverzeichnis.

92 Vgl. Überlegungen oben in Kap. 3 B.

1. Alle Softwarebestandteile, die bei der urheberrechtlichen Beurteilung gesondert betrachtet werden, sind für die Zwecke einer nutzbringenden Verwendung eine Einheit. Eine gebrauchsfähige EDV-Applikation besteht in der Regel aus einem umfangreichen System von begleitendem Dokumentationsmaterial, Dateien und Programmen. Nur diese Einheit von Softwarebestandteilen kann einem Dritten einen grösseren Nutzen bringen und ihm die als Grundlage für jede Verwertung notwendigen und umfassenden Kenntnisse über das ganze Konzept vermitteln. Einzelne Programme oder Dokumentationsteile können wertvolle Informationen enthalten, entbinden den Dritten aber nicht von der Mühe, für die Verwendung der Leistung aufwendige Studien über Anschlusspunkte anzustellen und die Software mit erheblichen Koordinationsrisiken durch eigene Bestandteile zu komplettieren. Nur in Ausnahmefällen kann die Übernahme einzelner ungeschützter Softwarebestandteile interessanter sein als der Erwerb einer ordentlichen Software-Lizenz, weil die Verschmelzung von fremden und eigenen Softwarebestandteilen ein sehr aufwendiges und risikoreiches Unterfangen ist⁹³. Lücken können daher in Kauf genommen werden. Die fehlende Schutzfähigkeit von einzelnen Softwarebestandteilen, insbesondere von kleineren Programmen und Steuerangaben, kann die Tauglichkeit des urheberrechtlichen Systems nicht in Frage stellen. Es reicht aus, wenn der Schutz der im Rahmen einer gebrauchsfähigen Applikation regelmässig vorhandenen umfangreicheren Softwarebestandteile gewährleistet ist. Kleinere ungeschützte Applikationsteile sind meistens von geringem Wert.

2. Gleiche Überlegungen sind auf der Ebene der einzelnen Softwarebestandteile, insbesondere der Programme, anzustellen. Selbst wenn in einem Programm die ungeschützten Elemente in einem wesentlich geringeren Umfang vertreten sind als die geschützten, zwingt die komplizierte Verwobenheit dieser Elemente zu einer aufwendigen Analyse, Kontrolle und Umarbeitung des ganzen Programmes. Deshalb ist auch bei einer minimalen individuellen Färbung das Abstandhalten mit einem beträchtlichen Aufwand verbunden. Von einem Verblässen der bescheidenen individuellen Züge und einer freien Benutzung der Programmvorlage kann erst die Rede sein, wenn die geschützten Teile auf eine andere Weise mit den ungeschützten verknüpft sind. Der Nachschaffende ist also gezwungen, die Gedankengänge des Originalschöpfers bis in alle Einzelheiten nachzuvollziehen. Das trifft vor allem für Änderungen im Bereich der Programmstruktur (innere Form) zu. Der Schutz von wenigen Elementen hat eine Ausstrahlung, die deutlich über den beschränkten Schutzbereich hinausgeht, weil der ganze inhaltliche Bereich in die Umarbeitung einbezogen werden muss.

3. Schliesslich werden die Schutzwirkungen noch durch die Bedeutung der Formgebung verstärkt. Das nicht schützbares Konzept, das Lösungsprinzip, steht bei Programmen im Vordergrund und hat einen bedeutenden wertmässigen Anteil an der Software. Die konkrete und teilweise geschützte Gestaltung hat nur eine untergeordnete, dienende Funktion. Für die Software gilt mit etwas anderen Gewichtungen das gleiche wie für andere Verstandeswerke. Wie in Kap. 2 E III hingewiesen, darf der für die Formgebungsaufgabe aufgewendete wertmässige Anteil an der Gesamtleistung jedoch nicht unterschätzt werden⁹⁴. Die Kodierung, d.h. das Schreiben des Programmes nach den genau vorgegebenen Lösungsprinzipien, die Befolgung der zahlreichen Regeln im Formalen mit den vielen Fehlermöglichkeiten, die Erfassung (Übertragung auf magnetische Datenträger), die maschinellen Umwandlungen, die Prüfung und Korrektur der zahlreichen Regelverstösse und andere mit der Form zusammenhängende Tätigkeiten, verursachen erhebliche Kosten. Diese Tätigkeiten wären auch dann erforderlich, wenn ihnen die ungeschützten wesentlich inhaltlichen Elemente einer fremden Softwarelösung zugrundegelegt werden könnten. Der im Bereich der Form sich auswirkende urheberrechtliche Schutz zwingt – etwas vereinfachend – zu einem weit aufwendigeren «Umweg» für die Nutzbarmachung von frei verfügbaren Kenntnissen, als dies der Fall ist, wenn die im Werk enthaltenen Kenntnisse, losgelöst von einer sehr qualifizierten Form, einen unmittelbaren abstrakten Nutzwert haben oder wenn diese nach einem weniger aufwendigen, flexibleren Umformungsvorgang (z.B. durch Verwendung der Umgangssprache) in einem neuen nutzungsfähigen Medium verwertet werden können. Der bescheidene urheberrechtliche Schutz entfaltet seine beträchtlichen Schutzwirkungen in einem Bereich, der für die Realisierung der Gesamtleistung nicht unbedeutend ist. Auch bei der Einhaltung eines minimalen Abstandes von der Vorlage muss wegen der «starr» Beschaffenheit des Ausdrucksmittels und der arbeitsintensiven Herstellungsprozeduren mit grossen Investitionen gerechnet werden.

Im folgenden ist der Schutzbereich und die Schutzwirkung bei den Softwarebestandteilen im einzelnen zu würdigen. Dabei soll wiederum versucht werden, entsprechend den beiden bekannten Antithesen zu differenzieren.

II. *Formschutz*

Wie dargelegt, können bei allen Softwarebestandteilen im Bereich der Form individuelle Elemente nachgewiesen werden. Ungeachtet der ver-

94 Entsprechende Studien existieren nicht. Der Verfasser legt diesen Aussagen die in Kap. 2 E III aufgeführten Zitate und eigene mehrjährige Erfahrungen zugrunde.

schiedenen Lehrmeinungen ist die einfache Kopie die untere Verbotsgrenze. Der Schutz geht aber weiter. Die Abänderung von wenigen Programmschritten, sprachlichen Formulierungen und sonstigen darstellerischen Elementen genügt ebenso wie die Übersetzung in eine andere Programmiersprache, Neukompilation oder vollständige, im Aufbau jedoch eng angelehnte Umgestaltung des blossen Sprachkleides nicht. Der meist vorhandene individuelle Plan der Gedanken, die innere Form des Ausdrucksmittels, zwingt zur Einhaltung eines grösseren Abstandes. Die frei variierbare Gliederung des Stoffes bzw. die innere Anordnung im jeweils verwendeten Ausdrucksmittel muss so stark vom Vorbild abweichen, dass die ursprünglichen Züge verblassen und von einer unzulässigen Übernahme, einer über den Bereich des frei verfügbaren wesentlichen Inhaltes hinausgehenden Aneignung fremden schöpferischen Gutes, nicht mehr gesprochen werden kann.

Weil umgangssprachliche Konzept- und Programmbeschreibungen, System- und Bedienungshandbücher, Flussdiagramme und Programme aus verschiedenen Gründen nach einer einheitlichen Entwicklungslinie aufeinander abgestimmt werden müssen und dem gleichen, teilweise individuellen Gedankenaufbau folgen, zwingt in der Regel die Umarbeitung der einen Teile auch zur Änderung der anderen. Es ist ohne Zustimmung des Rechteinhabers unzulässig, z.B. in enger Anlehnung an das Dokumentationsmaterial früherer Programmierungsphasen (Konzept-, Programmbeschreibungen, Flussdiagramme usw.) Programme in irgendeiner Programmiersprache herzustellen. Die damit verbundenen Übernahmen oder Umgestaltungen von formalen Elementen (Daten- und Datensatzanordnungen, Merkmale, Texte, Bildschirm- und Druckbilder, geschützte inhaltliche Strukturen usw.) verletzen die Rechte der Urheber. Die Rechte an den verwendeten Werken ermöglichen es, die Benutzung bzw. die Verbreitung der abhängig geschaffenen Programme zu verhindern. Das gilt auch dann, wenn der blossen Umgestaltung noch persönliche Beiträge des Zweitschaffenden hinzugefügt wurden. Zulässig ist nur die Aneignung der Lösungsideen, ohne dass die individuellen Ausdrucksmittel und gedanklichen Anordnungen des zugrundegelegten Werkes erscheinen⁹⁵. Der Formschutz ist beschränkt, jedoch sehr wirksam, weil er insbesondere bei Programmen zu einer kostspieligen Analyse und Umarbeitung der ganzen Form zwingt⁹⁶.

95 Es genügt demnach z.B. nicht, wenn Anordnungen und umgangssprachliche Texte für System- und Fehlermeldungen, Anordnungen und Bezeichnungen von Programm- und Datenelementen, Programmroutinen usw. einem Systemhandbuch oder Vorlageprogramm entnommen werden und im abhängigen Sourceprogramm oder in speziellen Textdateien mit gleichem oder ähnlichem Wortlaut oder vergleichbarer Anordnung als Textkonstanten oder Datenfelder neu definiert werden.

96 MOOERS C. N., S. 59 und oben Abs. I.

Probleme ergeben sich wegen der Unklarheit des Hilfsbegriffes der inneren Form. Einerseits kann von schützbaeren Elementen im Bereich der inneren Form der wirksamste Schutzbeitrag erwartet werden, weil ihr Schutz umfassendere Umarbeitungen erfordert, als sie mit der Unterstützung von Hilfsprogrammen⁹⁷ ausgeführt werden können. Andererseits verunmöglicht die Unschärfe dieses Hilfsbegriffes, die geschützten Zonen in den einzelnen Softwarebestandteilen genauer zu lokalisieren.

III. *Einbezug des Schutzes von nicht gemeinfreien inhaltlichen Elementen*

Die Schutzqualität wird ebenfalls entscheidend durch die erforderliche schwierige Abgrenzung der geschützten und gemeinfreien Bereiche⁹⁸ bestimmt. Einigermassen praktikable Ansätze könnten sich in der folgenden Richtung anbieten:

Die Softwareschöpfungen enthalten stets zahlreiche, mehr oder weniger in sich geschlossene Lösungselemente auf verschiedenen Ebenen. Als Lösungselemente können die einer Informationseinheit, einer logischen Einheit von Daten oder Zeichen, zugehörigen Umformungsregeln angesehen werden, die sich oft weiter in eine Vielzahl von selbständigen Unterelementen aufteilen lassen. Für die Lösungselemente verschiedener Stufen bieten sich meistens alternative Lösungswege an. In einem Lagerbewirtschaftungsprogramm ist z.B. unter vielen anderen das Lösungselement Lagerwertberechnung enthalten, welches logisch auf mehreren Datenelementen (Mengenangaben, Preise) basiert. Für die Ermittlung des Lagerwertes bestehen bereits von den betriebswirtschaftlichen Grundlagen her mehrere Berechnungsvarianten. Die vom EDV-Anwender getroffenen Auswahlen von Berechnungsvarianten gepaart mit den für jede Variante gewählten äquivalenten Lösungsmöglichkeiten bis hin zu den konkreten Anordnungen auf den Ein- und Ausgabeträgern lassen vielfältige Bezüge entstehen, die eine eigenartige oder zumindest sehr ausgefallene Struktur aufweisen. Eine kleinere Zahl von Lösungselementen ist auch als ganze Struktur freizuhalten. Anders als bei kleineren Strukturen mit beschränkteren Gestaltungsspielräumen können die Gemeininteressen nicht mehr beeinträchtigt werden, wenn bei einer grösseren Zahl von Lösungselementen und -wegen das sich ergebende Gewebe von Kombinationen oder Anordnungen und ihre Verbindungen zu bestimmten Auslösesymbolen im Bereich des Ausdrucksmittels geschützt wird. Abstandhalten ist in diesen Fällen ziemlich aufwendig, jedoch

97 Es handelt sich um Datei-Bearbeiter oder Text-Editoren, die für die Behandlung von Sourceprogrammen von den meisten Herstellern zur Verfügung gestellt werden.
98 Allgemeines oben Kap. 5 B II, vgl. auch ULMER E., Wiss. Werke, S. 15.

nicht schwierig, weil auf zahlreiche alternative Anordnungsmöglichkeiten ausgewichen werden kann. Einzelne Lösungselemente, d.h. wesentliche in sich geschlossene inhaltliche Teile der Software werden dadurch nicht blockiert, weil lediglich gefordert wird, dass die Anordnung der Lösungselemente und ihre Beziehung zu bestimmten Merkmalen und Symbolen abgeändert wird, vor allem teilweise frei variierbare Daten-, Datei-, Programm- und Befehlsstrukturen oder deren Bezeichnungen umgestaltet werden. Dieses Erfordernis kann selbst die denkbar abhängigste Art der inhaltlichen Nachschöpfung, die Übernahme sämtlicher Lösungselemente einer Vorlage, nicht behindern. Es bleibt gewährleistet, dass Programme mit gleicher Verarbeitungswirkung und ähnlicher Leistungsfähigkeit hergestellt werden können. Selbst der «Lösungsmix» einer ganzen EDV-Anwendung wird dadurch nicht monopolisiert.

Wie bereits in Kap. 5 B II 3 ausgeführt, ist es mit den Interessen der Allgemeinheit zu vereinbaren, dass auch formbestimmende Anweisungen⁹⁹ eines Programmes urheberrechtlich geschützt werden. Bei den vom Programm gesteuerten Anordnungen der Daten auf den Ein- und Ausgabegeräten bestehen viele Wahlmöglichkeiten. Mitteilungs- bzw. Darstellungsmonopole können nicht entstehen. Ebenso unbedenklich ist es deshalb, die individuellen Elemente von Bildschirm- und Druckbildern direkt zu schützen¹⁰⁰.

Auch der Umfang des schützbaren inhaltlichen Bereiches ist sehr klein. Die Schutzwirkung ist wegen des zu erwartenden Änderungsaufwandes auch unter diesem Gesichtspunkt gross. Sie reicht in den überwiegenden Fällen aus, dass bei gleichen Anforderungen an EDV-Lösungen der Weg zum Schutzrechtsinhaber vorgezogen wird.

IV. Nebenwirkungen des Urberschutzes auf Programmier- und Betriebssystemsprachen, Betriebssystemtechniken und Interpretationsprogramme

In Kürze ist auf eine von MOOERS¹⁰¹ erläuterte, etwas spekulative, im wesentlichen aber durchaus mögliche Nebenwirkung des Urberschutzes hinzuweisen.

Der Schutz der konkreten Form von Softwarewerken kann eine mittelbare Schutzwirkung ausüben auf urheberrechtlich nicht erfassbare ab-

99 Programminstruktionen, die Datenanordnungen auf den Ein- und Ausgabeträgern bestimmen, namentlich Listen-, Bildschirm- und sonstige Datendarstellungen steuern. Ferner auch Verzweigungsanweisungen, die Übergänge innerhalb der freigewählten Programmstruktur sicherstellen.

100 Kap. 6 B II 2 oben.

101 MOOERS C. N., (Rockford Research Inc., Cambridge, Massachusetts), S. 59ff. Dieser Autor bejaht den Softwareschutz auf einer ähnlichen Grundlage wie der in dieser Arbeit dargestellten.

strakte Dinge, z.B. auf hochentwickelte Sprachen von Programmierungs- und Betriebssystemen, Datenbank-Managementsystemen, Interpretations- und Generatorsystemen. Letztlich kann aus denselben Überlegungen gar das Kommandosystem einer ganzen Computerfamilie geschützt und anderen Computerherstellern verwehrt werden, durch die verhältnismässig einfache Herstellung von Mikroprogrammen eine vollständige Verträglichkeit mit einer Computerfamilie herzustellen.

In jedes mit sprachlichen Mitteln anzuwendende Softwaresystem sind zwei verschiedene Arten von Texten involviert, die nach den vorne dargestellten Grundsätzen in ihrer konkreten Form Urheberrecht genießen. Zum einen handelt es sich um den vorwiegend umgangssprachlichen Text, der die zahlreichen Details einer Sprachstruktur, die ihr zugehörigen Symbole oder Merkmale und die mit diesen verknüpften Funktionen sowie zahlreiche andere Konventionen definiert. Zum anderen besteht ein Programm, z.B. ein Compilerprogramm, das die nach dieser Beschreibung hergestellten, in diesem Zusammenhang nicht weiter interessierenden (Anwender-) Programme interpretiert und für den Gebrauch mit entsprechenden Funktionsmerkmalen ausstattet. Es liegt auf der Hand, dass zwischen dem in einem Text definierten Sprachsystem und dem diese Sprache verarbeitenden Programm eine ganz genaue Übereinstimmung, eine «mirror-picture-relationship», bestehen muss. Es entstehen für ein Softwaresystem folgende derartigen Paarbeziehungen von geschützten Texten:

«Mirror Pairs of Texts»¹⁰²

Defining Text for:	Program Text for:
Job Control (Betriebssystem) Language	Operating System (Betriebssystemprogramm)
File Interrogation Commands	Information Retrieval System
Programming Language Statements	Compiler for the Language
Interactive System Commands	Interpreter for an Interactive System.»

Jede auf die Benützung dieser Sprachsysteme und ihrer funktionellen Merkmale zielende Handlung muss wegen der erforderlichen Übereinstimmung exakt auf einer dieser Texte bzw. auf die in ihnen enthaltenen, in vielen Fällen auf einer freien Auswahl beruhenden Symbole, Textstrukturen, Funktions- und Symbolverknüpfungen, Merkmalssequenzen und vielen anderen korrespondierenden Details aufbauen. Wer also ein Compiler-, Betriebssystem- oder Mikroprogramm entwickelt, muss sich genau an die beschriebenen Definitionen anlehnen. Wer umgekehrt ein entsprechendes Programm als Vorlage verwendet und damit das ganze System rekonstruiert,

102 Im folgenden zitiert nach MOOERS C. N., S. 61.

muss ebenfalls sehr genau alle Merkmale der Vorlage in sein abhängiges Programm und die sie begleitenden Definitionsbeschreibungen übernehmen. Aus jeder Herstellung eines paarigen Textes muss sich wegen der sachlich bedingten engen Anlehnung eine Urheberrechtsverletzung ergeben. Der abhängige Text ist eine Bearbeitung oder ein Werk zweiter Hand, weil der Nachschöpfer mit grosser Wahrscheinlichkeit auch Eigenheiten übernehmen muss, die nicht bloss Teil des gemeinfreien Inhaltes, sondern als die schutzzfähigen, weil einer freigewählten sprachlichen Konvention folgenden Ausdrucksmittel eines Systems mit logischen Funktionseinheiten anzusehen sind.

MOOERS¹⁰³ stellt derartige textliche Umwandlungen mit Übersetzungen oder Bearbeitungen gleich. Er vergleicht mit der Schaffung eines dramatischen Werkes, das in der Struktur des Handlungsablaufes und der Umstände genauestens einer Novelle nachgebildet wurde. Auch wenn man sich wegen des andersartigen Charakters von Softwarewerken Zurückhaltung aufzuerlegen hat bei der Festlegung des Schutzbereiches, kann man sich der hier wiedergegebenen Auffassung nicht verschliessen, dass die notwendige starke Abhängigkeit der sachlich in Beziehung stehenden Texte nicht oder nur mit grosser Mühe Raum für eine eigenständige Darstellung bietet und daher gleichzeitig eine Art von faktischer Mitteilungs- und Anwendungsexklusivität für ein Sprach- und darauf aufbauendes Softwaresystem entsteht.

Diesen Wirkungen unterstehen selbstverständlich die folgenden Sachverhalte nicht: Es ist zulässig, die generellen Ideen eines Sprach- und Softwaresystems zu übernehmen und in eine selbständige Darstellung einfließen zu lassen. Das dürfte aber zur Folge haben, dass keine Kompatibilität auf Systemebene mehr besteht. Es ist ferner zulässig, die geschützten Texte zu gebrauchen, um nach ihren Anweisungen selbständige Programm- oder andere Befehlsstrukturen herzustellen, weil sich diese Verwendungsart weitgehend von der konkreten Vorlage löst.

Bei den gängigen Programmiersprachen sprechen, soweit sie nicht bereits wegen ihrer allgemeinen Verbreitung zum Gemeingut gehören, zu viele Interessen von allen beteiligten Kreisen für eine ungehinderte Verbreitung. Es ist unwahrscheinlich, dass die hier bezeichneten Schutzmöglichkeiten ernsthaft in Betracht gezogen werden könnten. Nicht auszuschliessen, jedoch auf sehr unsicherem Boden sind Bestrebungen, durch derartige Nebenwirkungen des Urheberschutzes der Anwendung von typischen Sprachmustern gewisser Computerfamilien, Generator- oder anderen Softwaresystemen entgegenzutreten.

V. *Zur gerichtlichen Durchsetzung des urheberrechtlichen Softwareschutzes*

Der Urheber hat alle anspruchsbegründenden Umstände darzutun. Dazu gehören insbesondere der Nachweis der Werkeigenschaft und der Beweis der Verletzungstatbestände. Beim Verletzungsbeweis ist darzulegen, dass die dem Urheber vorbehaltenen individuellen Bereiche einer Softwareerschöpfung verwendet wurden¹⁰⁴.

1. *Nachweis der Werkeigenschaft*

Der Nachweis der Werkeigenschaft ist bei den meisten Ergebnissen der Softwareherstellung nicht sehr schwierig. Gelegentlich dürfte es nicht zu umgehen sein, hierfür Sachverständige beizuziehen.

Unproblematisch ist der Nachweis bei Softwareergebnissen mit umgangssprachlichen Ausdrucksmitteln und mit grafischen oder schematischen Darstellungen. Deshalb dürfte auch die etwas schwierigere Beurteilung der Programme weiteren Personenkreisen möglich sein, weil für deren Bewertung meist einfach verständliche sekundäre Beurteilungsgrundlagen zur Verfügung stehen. Anhand der durch die Programme erzeugten Ergebnisse, vor allem Bildschirm- und Druckbilder, und anhand von Unterlagen früherer Entwicklungsphasen können die in den Programmen enthaltenen schöpferischen Elemente auf eine allgemeinverständliche Weise dargelegt werden. In der Regel dürfte eine punktuelle Betrachtung genügen¹⁰⁵, wenn man gleichzeitig fordert, dass die zu beurteilenden Softwarebestandteile einen gewissen Umfang aufweisen. Genügen dürften bei Programmen im allgemeinen, wie KOLLE¹⁰⁶ vorschlägt, etwa fünfhundert Programmzeilen.

2. *Beweis der Verletzungstatbestände*

Wie beinahe einhellig (für alle infragekommenden Schutzformen) festgestellt wird, ist es bei der Software sehr schwierig, Verletzungstatbestände aufzudecken¹⁰⁷. Die wichtigsten Softwarebestandteile sind für die Zwecke des normalen Gebrauchs magnetisch aufgezeichnet und deshalb im Zusammenhang mit dem regelmässigen praktischen Einsatz nicht unmittelbar wahrnehmbar. Oft ist die genauere Eigenart der Programme nicht ein-

104 Dazu oben Kap. 6 C I.

105 Zutreffend u.a. KÖHLER R., S. 86.

106 KOLLE G., GRUR 1974, S. 9.

107 Einzelne Ausführungen bei KOLLE G., GRUR 1974, S. 616f; KÖHLER R., S. 35ff; THURMANN R. V., GRUR Int 1969, S. 217f; SCHWAIGER/KOCKLER, S. 365 und dortige Hinweise; BRAUN B., BB 1971, S. 1347.

mal den Anwendern und schon gar nicht dem Bedienungspersonal bekannt. Nur eine sorgfältige und fachlich anspruchsvolle Überprüfung anhand von Programmlisten lässt schlüssige Beweise von Verletzungen zu. Die Gefahr im Verborgenen bleibender Verletzungen ist bedeutend geringer, sofern Software unbefugt geschäftsmässig angeboten wird, da die begleitende Werbung gewisse kennzeichnende Merkmale allgemein bekannt werden lässt. Die im allgemeinen in den Werbematerialien veröffentlichten Beispiele von Bildschirm- und Listenbildern könnten sogar direkt auf Urheberrechtsverletzungen hinweisen. Etwas einfacher ist in der letzten Zeit die Aufdeckung von Verletzungen deshalb geworden, weil die immer mehr verbreiteten Dialogprogramme (Ein- und Ausgabe auf Bildschirmen), ähnlich Programmen mit Ein- und Ausgaben auf Druckern, ihre besonderen Eigenschaften bei jedem Gebrauch dem Bedienungspersonal und weiteren, teilweise sogar den ausserhalb der betrieblichen Organisation stehenden Personen wie Servicetechnikern, Adressaten von Computeroutputen usw. offenbaren. Im Gegensatz zu den noch immer sehr gebräuchlichen Stapel- oder Schubverarbeitungssystemen bleibt das Programm bei Dialogsystemen aussenstehenden Betrachtern nicht mehr weitgehend verborgen, weil es nicht nur intern abläuft. Es bleibt im ganzen gesehen jedoch dabei, dass der Rechtsinhaber meistens geringe Chancen hat, einen Verletzer aufzuspüren. Zu unterscheiden von den Problemen, eine Rechtsverletzung überhaupt aufzudecken und die notwendigen Vergleichsunterlagen beizubringen, ist die Aufgabe, anhand der Gegenüberstellung des geschützten und verletzenden Programmes im einzelnen die Abgrenzung zwischen freier Benützung und unerlaubter Bearbeitung oder Umgestaltung des geschützten individuellen Bereiches vorzunehmen. Diese Aufgabe dürfte bei geringer Selbständigkeit des verletzenden Programmes mit Hilfe von Fachleuten ohne weiteres zu bewältigen sein. Die Analyse könnte aber bei umfangreicheren oder verschleiernenden Änderungen sehr kompliziert und aufwendig werden. Es ist auch sehr zu bezweifeln, dass solche Analysen noch justitiabel sind. Mögliche Behelfe wären Beweiserleichterungen oder präventive Massnahmen für Beweis Zwecke, z.B. Einbau von Code-Wörtern oder extrem ausgefallenen, teilweise inaktivierten Befehls-Redundanzen in Programmen¹⁰⁸.

Schwierigkeiten bei der Aufdeckung und beim Nachweis von Verletzungen bestehen bei allen Schutzformen in ähnlichem Masse. Wegen derartiger Schwierigkeiten kann das Urheberrecht nicht unvorteilhafter erscheinen als die anderen Schutzsysteme, insbesondere auch die vorgeschlagenen Sonderschutzsysteme¹⁰⁹.

108 Dazu z.B. KOLLE G., GRUR 1973, S. 617.

109 Vgl. Kap. 7 unten. Falls Sonderschutzsysteme gegebenenfalls den Verletzungsnachweis in nicht besonders qualifizierten, nicht-individuellen Bereichen zuliesse, müsste man mit der ebenso schwer verifizierbaren Einwendung rechnen, dass es sich um Pa-

Schwere Bedenken bezüglich der Wirksamkeit eines rechtlichen Schutzes sind generell am Platze¹¹⁰. Präventiven technischen Schutzmassnahmen unter dem Gesichtspunkt der Datensicherung¹¹¹ wird deshalb auch bei einer Vereinfachung des Rechtsschutzes schon zu Beweis Zwecken stets eine wesentlich grössere Bedeutung beigemessen werden müssen. Wie bereits vorne festgestellt, bestehen in beinahe allen Bereichen der Softwareherstellung wirksame technische Schutzmöglichkeiten. Die Chancen, durch Fremdverwertung Erträge zu erzielen, werden dabei kaum beeinträchtigt. Dem rechtlichen Schutz kann unter Praktikabilitätsgesichtspunkten stets nur eine sehr subsidiäre Funktion zukommen. Erwartungen dürften nicht begründet sein, dass von einer geeigneteren Form eines durch internationale Abkommen unterstützten Rechtsschutzes Impulse für eine stärkere Verbreitung und grosszügigere Offenbarung der Software ausgehen können und insbesondere auch «den Entwicklungsländern der Zugang zu Informationen über Computersoftware erleichtert werden kann»¹¹².

F. Verschiedene Fragenkreise zum Urheberschutz der Software

Im folgenden werden Fragen behandelt, die nicht in den Problemkreis «Werk» gehören. Berücksichtigt werden vor allem Argumente, die in der neueren schweizerischen Literatur¹¹³ vorgebracht wurden.

parallelentwicklungen handle. Im übrigen kann auch der nicht-individuelle Bereich derart verschleiert werden, dass ähnliche Beweisschwierigkeiten nicht auszuschliessen sind. Zurückzuhalten ist auch mit den Erwartungen bei der Einrichtung eines Hinterlegungssystems de lege ferenda. Die Nachteile einer früheren oder späteren Offenbarung, die als Gegenleistung für die Gewährung eines umfassenderen Sonderschutzes in Betracht zu ziehen wäre, könnten durch die gegenüberstehenden Vorteile bei der Lizenzierung und Identifizierung kaum aufgewogen werden. Es sprechen mehrere andere praktische Gründe gegen ein Hinterlegungs- oder Registrierungssystem. Weitergehende formelle Schutzvoraussetzungen würden neben der Belastung der grösstenteils kleineren Softwareunternehmen und des Staates schon deshalb erhebliche Probleme mit sich bringen, weil Programme – selbst längere Zeit im Einsatz stehende Standardprogramme – häufig geändert oder ergänzt werden müssen. Neben einer Konkurrenzierung durch den Staat entstünden auch Doppelspurigkeiten mit bereits existierenden speziellen Periodika, die Eigenschaften von Softwarekomponenten bekannt machen, um Kontakte für Lizenzen zu vermitteln (weiteres in den Materialien zu den WIPO-Mustervorschriften, GRUR Int. 1978, S. 288).

110 Ausgeprägt z.B. bei SCHWAIGER/KOCKLER, S. 361; BRAUN B., S. 1347.

111 Oben Kap. 3 B.

112 Materialien zu den WIPO-Mustervorschriften für den Schutz von Computersoftware, GRUR Int. 1978, S. 286ff. Die WIPO-Mustervorschriften sind u.a. den Zielsetzungen verpflichtet, neben dem erwähnten Anreiz zur Offenbarung auch die Harmonisierung und die Lückenfüllung der nationalen Gesetzgebungen zu bewirken. Näheres in Kap. 7.

113 Unter dem Gesichtspunkt «praktische Schwierigkeiten und Nachteile» vorgebrachte Argumente bei HEIM/CURCHOD, Schw. Mitt. 1972, S. 109ff; Ausführungen von PERRET F., ZSR 96 (1977), II, S. 276ff, in einem Hauptreferat anlässlich der Versammlung des Schweizerischen Juristenvereins 1977. Die von PERRET vorgelegte Arbeit zum weiteren Fragenkreis des Leistungsschutzes ist die zuletzt im schweiz. Schrifttum erschienene Stellungnahme zum urheberrechtlichen Softwareschutz.

I. *Die rechtliche Erfassung der Verwendungsformen von Softwarewerken*

PERRET¹¹⁴ führt in seiner Arbeit aus: «... la fixation du programme sur des cartes perforées ou sur tout autre support ne peut être considérée comme une reproduction selon la conception classique du droit d'auteur, car un tel enregistrement est destiné à la machine et n'a donc pas pour fonction de communiquer l'œuvre au public.» Zunächst ist festzustellen, dass gleich wie die Schallplatte, die Tonkassette, der Filmstreifen u.a. auch die Lochkarte oder andere Speichermedien (Magnetband, Magnetplatten usw.) Informationsträger sind, die in erster Linie für den Gebrauch auf einer Maschine oder auf einem Gerät geeignet und bestimmt sind. Das ist aber unerheblich, weil nach herrschender Auffassung¹¹⁵ jede Konkretisierung einer sich als Werk ausweisenden schöpferischen Leistung, auch wenn sie erst mit Hilfe einer technischen Einrichtung für den Menschen direkt wahrnehmbar gemacht werden kann, dem Urheberschutz untersteht. Ungeachtet dessen, dass der Inhalt der Lochkarte z.B. unmittelbar durch den Menschen wahrgenommen und auch verstanden werden kann, ist festzuhalten, dass andere magnetisch aufgezeichnete Softwarebestandteile ohne weiteres und jederzeit auch mit Druckern und Sichtgeräten in eine verständliche Ausdrucksform gebracht werden können. Während der Entwicklung und des Gebrauches (z.B. bei der Wartung und der Revision der Programme) geschieht das regelmässig. Das Urheberrecht verlangt vom schutzfähigen Werk nicht eine bestimmte und besonders qualifizierte Einwirkung auf die menschlichen Sinne¹¹⁶. Es ist unerheblich, ob die Werke primär den

114 PERRET F., ZSR 96 (1977), S. 275.

115 Die Lehre folgt nicht der amerikanischen «piano roll-Doktrin» (Fall White-Smith Publ. Co. v. Apollo Co., 209 U.S. 1, 1908). In den Herrschaftsbereich des Urhebers fallen auch Reproduktionsarten, die das Werk in einer den menschlichen Sinnen mittelbar wahrnehmbaren oder verständlichen Festlegung erfassen. Vgl. auch Zitate oben Kap. 2 E IV, Kap. 5 B II 5, HUBMANN H., 4. Aufl., S. 131f; DITTRICH R., S. 13f; GOTZEN F., DdA 1977, S. 19.

Ganz klar in diesem Sinne das neue amerikanische Urheberrechtsgesetz von 1976 in Art. 102 (a) in Verbindung mit Art. 101 (vgl. KOLLE G. in DdA 1977, S. 73). Namentlich gehören dazu die Einspeicherungen der Werke, auch Softwarewerke in einen Computer (näheres hierzu unter internat. Blickwinkel: WIPO, Report Working Group on Copyright Problems arising from the Use of Computers, Genf, Juni 1978, S. 3), die Vervielfältigungen auf Datenwischenträgern, computerinterne Vervielfältigungen auf Speichern oder Registern (WIPO-Publ. Nr. 814 (englische Fassung), Genf 1978, S. 5: Hier geäußerte Zweifel, ob solche computerinterne Vervielfältigungen für das Urheberrecht genügen, sind nicht am Platze, weil der Schutz in jedem Fall auch bei der stets erforderlichen erstmaligen Einspeicherung (Fixierung des Werkes durch manuelle Datenerfassung auf externen Datenträgern oder Übertragung von vorbereiteten Datenwischenträgern auf andere, einem bestimmten Computer angeschlossene externe Datenträger) eingreifen kann, die erst Grundlage für die interne Vervielfältigung ist), die Ausgabe des Werkes auf Papier, Archivfolien oder Sichtgeräten oder deren Vervielfältigung mit Geräten, die nicht Bestandteil der EDV-Anlage sind, z.B. Fotokopiereinrichtungen.

116 KOLLE G., GRUR 1974, S. 8 und dortige Zitate.

Zweck¹¹⁷ haben, einen maschinellen Ablauf zu steuern oder an ein beliebiges Publikum oder einen kleinen Kreis von eingeweihten Software-Fachleuten verbreitet zu werden¹¹⁸. Hängt doch die Schützbarkeit vom Verwendungszweck in keiner Weise ab¹¹⁹. Der Schutz hängt ebenfalls nicht von einer bestimmten gewerblichen Verwertbarkeit oder Verwertungsmöglichkeit ab¹²⁰. Geschützt sind auch Werke, die nicht zur Veröffentlichung bestimmt sind¹²¹ (z.B. Tagebücher, die ihr Verfasser geheimzuhalten beabsichtigt). Im weiteren fällt auch nicht in Betracht, ob die maschinenverständlichen Sprachelemente nur von wenigen Spezialisten oder vom breiten Publikum verstanden werden können¹²².

Schliesslich ist unwesentlich, mit welchen technischen Hilfsmitteln¹²³ und zu welchem Zweck eine bestimmte Vervielfältigung vor sich geht. PERRET¹²⁴ fährt fort: «En effet, l'introduction des cartes perforées dans l'ordinateur de même que leur mémorisation subséquente par ce dernier échapperaient en tout cas au droit exclusif puisqu'il s'agit là d'une utilisation du programme et non de sa reproduction.» Der urheberrechtliche Schutz hängt nicht von einer bestimmten gewerblichen Verwertungsart ab, sondern knüpft bei bestimmten gesetzlich eingeräumten Verwendungsbefugnissen an. Zweck der Vervielfältigung ist der Gebrauch auf einem Computer. Die Voraussetzung für jeden Gebrauch, eingeschlossen die Kompilation für die Erstellung von Objektprogrammen, ist eine Reproduktion, sei es, dass die Software dem Anwender als Kopie auf Vervielfältigungsstücken (Platten, Bänder usw.) zugänglich gemacht werden muss oder dass mit jedem einzelnen Gebrauch mehrere maschinelle Kopiervorgänge (Datenzwischenträger – peripheres Gerät – Zentralspeicher – Befehlsregister) verbunden sind. Der Gebrauch kann auch nicht bloss als eine Verwendung des freien Verfahrens bzw. des Inhaltes oder Konzeptes angesehen werden, weil auch der computermässige Gebrauch die Verwendung der konkreten Mitteilungsförm vor-

117 TROLLER A., IG I, S. 417: «Der Zweck, dem ein Werk dient, ist nach einem allgemein anerkannten urheberrechtlichen Grundsatz nicht zu beachten.» Deutlich VE II Art. 1 Abs. 2 URG: «Werke im Sinne dieses Gesetzes sind ohne Rücksicht auf Wert und Zweck Schöpfungen ...».

118 ULMER E., Wiss. Werke, S. 12/13; auch ULMER sieht kein Hindernis für einen urheberrechtlichen Schutz der Programme, weil sich ihre Befehle primär an eine Maschine wenden. Eingehend aus amerikanischer Sicht MOOERS C. N., S. 56f mit Hinweis auf die Entscheidung des Supreme Court, MOZUR V. STEIN, 347 U.S. 201 (1954): «We find nothing in the copyright statute to support the argument that the intended use in industry of an article eligible for copyright bars or invalidates its registration. We do not read such limitation into the copyright law».

119 KUMMER M., Werk, S. 51 und 200.

120 ULMER E., Verlagsrecht, S. 21.

121 ULMER E., Verlagsrecht, S. 21; DITTRICH R., S. 12.

122 Entgegen ZAHN R., GRUR 1978, S. 214, ist ein Programm einem Spezialisten in weiten Bereichen aus sich heraus verständlich (vgl. oben Kap. 6 C I 1, Kap. 6 C IV 2 und Kap. 5 B II 6).

123 Art. 12 Abs. 2 Ziff. 1 URG: «... das Werk durch irgendein Verfahren wiederzugeben.»

124 PERRET F., ZSR 96 (1977), S. 275.

aussetzt und miteinschliesst. Der urheberrechtliche Schutz mag möglicherweise von begrenztem Wert sein, weil er nur indirekt bei der Vervielfältigung der konkreten Form und nicht direkt beim Gebrauch einsetzen kann. Durch die Kontrolle der Vervielfältigungen können aber weitgehend die Gebrauchsmöglichkeiten¹²⁵ bestimmt werden, weil der Gebrauch nur durch die Kopie der konkreten Ausdrucksform vermittelt werden kann. Wie bereits erwähnt, ist es irrelevant, dass nicht direkt bei den mit dem Gebrauch zusammenhängenden, nicht exakt verifizierbaren internen Kopiervorgängen angeknüpft werden kann, weil bereits die notwendige Verbreitung auf Datenwischenträgern und die erstmalige Eingabe in einen fremden Computer oder sonstige mit dem Gebrauch notwendigerweise zusammenhängende Handlungen wie Programmwartung, Analyse etc. in den Anwendungsbe-
reich des Urheberrechtes fallen.

Der Schutz kann aber ohne weiteres dann direkt beim Gebrauch einsetzen, wenn bei diesem gleichzeitig Vervielfältigungen von individuellen Listen- und Bildschirmbildern oder sonstigen schutzfähigen Elementen erzeugt werden.

Schliesslich argumentiert PERRET¹²⁶: «... et même si l'on voulait passer outre à toutes ces objections (vgl. obige Zitate) on se heurtera toujours à l'obstacle difficilement surmontable à notre avis qu'une telle reproduction serait à des fins purement privées et partant ne tomberait pas sous le coup de la loi sur le droit d'auteur. Il est vrai que cette reproduction a lieu dans l'entreprise; toutefois, elle n'est pas destinée à ses membres, mais à l'ordinateur qui doit exécuter le programme.» Selbstverständlich fallen die Kopiervorgänge innerhalb des Computers oft in den privaten Gebrauch. Es kann indessen, wie bereits oben erwähnt, an anderen Verbreitungshandlungen angeknüpft werden. Unerheblich ist, inwieweit und ob der primäre «Werkgenuss» nur mit Hilfe eines Computers erfolgen kann. Verlangt wird ja nicht ein bestimmtes ästhetisches Erlebnis beim Genuss des Werkes. Vielmehr ist allein entscheidend, ob eine schöpferische Leistung für den Gebrauch ausserhalb eines zulässigen Personenkreises verbreitet wird. Dies ist in den folgenden Fällen möglich: Weitergabe von Software auf Datenwischenträgern (Vervielfältigungsstücke in Form von Lochkarten, Magnetbändern und -platten usw.) an Dritte (körperliche Wiedergaben) und unkör-

125 Dazu auch HEIM/CURCHOD, S. 109: «... soll sich der von den interessierten Kreisen angestrebte Schutz aber auch auf die Benützung des Programmes, die keine Vervielfältigung darstellt, beziehen.» Selbstverständlich will man im Ergebnis auch und vor allem die Benützung verhindern. Die Voraussetzung für die Benützung ist immer irgendeine Verbreitung. Es ist zu unterscheiden zwischen den Motiven für den Schutz und den zum Schutz gesetzlich eingeräumten Verwendungsbefugnissen.

126 PERRET F., ZSR 96 (1977), S. 275.

perliche Wiedergaben bei der Eingabe in fremde Computer; Kopiervorgänge innerhalb eines Computers und bei der Ausgabe auf externe Datenträger, wenn bei Teilnehmerrechensystemen Dritte von ihren Terminals aus solche Vorgänge veranlassen und fremde Software kopieren¹²⁷ oder wenn ein Rechenzentrum für Dritte Arbeiten ausführt und hierbei ebenfalls Software vervielfältigt. Ferner übersieht PERRET, dass mit dem Einsatz des Programmes meistens individuelle Elemente in Form von Druck- und Bildschirmbildern (Output) wiedergegeben werden, die von Personen eines Unternehmens wahrgenommen werden können. Dasselbe trifft auch zu, wenn die Programme von EDV-Fachleuten eines Betriebes analysiert, gewartet oder anderweitig installiert werden.

Im übrigen ist der private Kreis, wenn man so will, eine allgemeine «Schwäche» des Schutzes bei allgemein reproduzierbaren Gegenständen¹²⁸ und eine Grenzmauer, vor welcher auch Sonderrechte de lege ferenda, die wie die WIPO-Mustervorschriften in § 5¹²⁹ einen Gebrauchsschutz für Software vorsehen, haltmachen müssten. Eine freie Vervielfältigung zum persönlichen Gebrauch¹³⁰ ist nicht mehr gegeben, wenn in- oder ausserhalb der Datenverarbeitungsanlage des Werknutzungsberechtigten speziell oder in Verbindung mit einer Benützung, die immer eine Vervielfältigung der Software voraussetzt oder einschliesst, Kopien hergestellt werden, die nicht ausschliesslich eigenen Zwecken dienen.

Ein eigenständiger Verletzungstatbestand des unbefugten Gebrauchs ist entbehrlich, weil bei umfassender Würdigung der mit der Weitergabe von Software verbundenen Vorgänge und bei einer zeitgemässen Interpretation des Begriffs der Vervielfältigung auch der Gebrauch der Computersoftware im aufgezeigten Rahmen verhindert werden kann. Bei der Interpretation des Vervielfältigungsbegriffes ist zu berücksichtigen, dass bereits heute zahlreiche andere Werkarten ohne Softwarecharakter mit Hilfe von datenverarbeitenden Einrichtungen verwertet bzw. verbreitet werden¹³¹. Es

127 Ein anschauliches Beispiel bei DIERSTEIN/FIEDLER/SCHULZ, S. 275: Ein Benutzer eines Time-Sharing-Systems liess von seinem Terminal aus die Software eines fremden Teilnehmers für eigene Zwecke auf einen Lochkartenstanzer ausgeben. Das Kartenpaket (Software im Wert von rund sechs Millionen Dollar) wurde nur dank einer zufälligen Intervention des Bedienungspersonals nicht weitergegeben.

Weitere Hinweise auf die Problematik unter strafrechtlichen Gesichtspunkten bei FISCHER T., Computer-Kriminalität, Bern 1979, S. 23f.

128 Dazu PEDRAZZINI M., ZSR 96 II (1977), S. 108.

129 Oben Kap. 7.

130 Art. 22 URG; eingehendere und restriktivere Regelung in VE II Art. 29f URG.

131 Beispiel: Aufbereitung, Speicherung und Wiedergewinnung (Ausgabe auf Datenträgern, Sichtgeräten usw.) von anderen wissenschaftlichen Texten mit Datenverarbeitungsanlagen. Vgl. oben Kap. 2 B 2.

handelt sich allgemein um eine neue Werkvermittlungsart, die sich infolge der zunehmenden Verbreitung von Computeranlagen im Verkehrsleben allmählich durchsetzen wird.

II. *Die Rechtszuordnung*¹³²

Dazu HEIM/CURCHOD: «Gegen einen Urheberrechtsschutz sprechen folgende Schwierigkeiten und Nachteile: ... (Aufzählung in Abs. II) ... Als Indiz für die Problematik des urheberrechtlichen Schutzes wird angesehen, dass sich der Urheber eines Programmes in vielen Fällen nicht identifizieren lässt, da die meisten Programme das Resultat der Zusammenarbeit einer ganzen Anzahl von Fachleuten sind.»¹³³ Im Tatsächlichen trifft das zu. Dieses Problem ist bei der Software aber nicht singulär. Es besteht weit über den Bereich des Film- und Fernsehschaffens hinaus in der gesamten Gegenwartskunst, «die gelegentlich bewusst der Entpersönlichung verpflichtet ist¹³⁴.» Mit der Metamorphose des Kunsterzeugnisses zur Ware ist in vielen Bereichen der Einzelhersteller dem Betrieb mit einer unbestimmten und keineswegs stabilen Personenvielzahl gewichen¹³⁵. Das hier angesprochene Problem ist ein allgemeines. Die traditionelle Urheberrechtszuordnung vermag generell nicht mehr den bestehenden Realitäten im Kunstschaffen zu genügen. Das Problem lässt sich allerdings kaum lösen, wie HEIM/CURCHOD anzunehmen scheinen, indem man die Definition des Rechtsobjektes aufgrund der zu erwartenden «Schwierigkeiten» von Fall zu Fall modifiziert¹³⁶, weil das zur Folge haben müsste, dass auch das grösstenteils entpersönlichte moderne Kunstschaffen entgegen einem objektiv-geltungszeitlichen Verständnis von Sinn und Zweck der geltenden Urheberrechtsbestimmungen vom Schutz auszuschliessen wäre¹³⁷. Die Fragenkreise Werk und Rechtserwerb sind strikte zu trennen.

132 Zur vorgeschlagenen Regelung in den WIPO-Mustervorschriften vgl. Kap. 7 B II unten.

133 HEIM/CURCHOD, S. 109; vgl. auch ZAHN R., GRUR 1978, S. 213; SCHWAIGER/KOCKLER, S. 365.

134 LARESE W., in Film und Recht 1978, S. 74.

135 LARESE W., eingehend in UFITA Bd. 79 (1977), S. 153ff = Festschrift SAMSON B., 1978, S. 153ff; Lösungsvorschläge (Zeichnung als Urheber) in Film und Recht 1978, S. 74ff und S. 101ff.

136 Wie LARESE (vgl. Fn 135 oben) zutreffend bemerkt, wären gerade wegen derartiger Probleme bei der Rechtszuordnung in den laufenden Urheberrechtsrevisionen noch neue Akzente zu setzen.

137 HEIM, Statist. Einmaligkeit, S. 66f, zum Schutzkriterium «eigenartige Schöpfung» von Art. 1 Abs. 2 URG. Es wird in dieser Arbeit ferner festgestellt, dass eine «weitere Fassung» des Werkbegriffes im KUMMERSCHEN Sinne mit dem geltenden Recht zu vereinbaren ist.

HEIM/CURCHOD stellen in Übereinstimmung mit der ausländischen Literatur fest: «Die übliche Dauer des Urheberschutzes erscheint für Computerprogramme zu lang (... 50 Jahre nach dem Tode des Urhebers). Zudem könnte durch Bearbeitung der Schutz beliebig verlängert werden. Selbst die interessierten Kreise sprechen sich gegen eine derart lange Schutzfrist aus.»¹³⁸ Nach dem bisherigen Gang der Entwicklung kann davon ausgegangen werden, dass die Nutzungsdauer eines Programmes auch bei wiederholter Bearbeitung eher ausnahmsweise wesentlich länger als fünfzehn Jahre ist¹³⁹. Das gibt in zweierlei Hinsicht zu Bedenken Anlass. Einmal ist der Gesichtspunkt der Verhältnismässigkeit hervorzuheben, weil die Schutzwirkung gemessen an der Nutzungsdauer, dem wahrscheinlichen zeitlichen Schutzbedürfnis, viel zu lange andauert¹⁴⁰. Das lässt sich, ebenso wie bei einem Grossteil der «Dutzendware» im heutigen urheberrechtlich geschützten Schaffen, nicht vermeiden, solange an einer einheitlichen Schutzdauer festgehalten wird und wohl auch werden muss, wenn nicht komplizierte, die Rechtssicherheit beeinträchtigende Abgrenzungsprobleme hervorgerufen werden wollen. Weniger harmlos sind andere Bedenken. Eine derart langandauernde Ausschlusswirkung ist nicht erwünscht bei Werken, die in erster Linie zum Gebrauch für nützliche Zwecke bestimmt sind. Das gilt für die Software in gleichem Masse wie für andere Werkkategorien wissenschaftlicher Art. Die Möglichkeiten einer ungehinderten Verbreitung bleiben allzu lange beschränkt, der Benutzerkreis kann in einem unerwünschten Masse exklusiv bleiben. «Patentähnliche» Überlegungen wären hier eher am Platz, weil die Software wie viele andere wissenschaftlichen Werke den Ursprung weniger im Urheber als in einem allgemeinen Stand des Wissens haben¹⁴¹. Die hier anzustellenden Betrachtungen sollen beschränkt bleiben auf den Hinweis, dass es sich hier wiederum nicht um eine typische, wenn sich auch hier in aller Deutlichkeit zeigende Problematik der Software handelt. Es ist wiederum zu bezweifeln, ob eine Differenzierung der Schutzdauer eine bessere Lösung bringen könnte. Tatsache bleibt, dass bei vielen wissenschaftlichen Werken der geringe Grad an individueller Prägung durch das Urheber-

138 HEIM/CURCHOD, S. 109.

139 Grund: technischer Fortschritt, Hardware- und Softwareverbesserungen, neue Lösungsideen und Bedürfnisse. Die «Bearbeitungsfähigkeit» von Programmen ist im übrigen beschränkt, weil nach mehreren Änderungen in der Regel ein nicht mehr rational nutzbares, unübersichtliches «Flickwerk» entsteht.

140 Von einer Art «Überpatent» zu sprechen (vgl. MOSER H., GRUR 1967, S. 639), ist aber unrichtig, weil der Patentschutz inhaltliche Elemente (Verfahrensregeln) erfassen würde, die im Urheberrecht keinen Platz haben.

141 Vgl. Kommentar zu den WIPO-Mustervorschriften, WIPO-Publ. No. 814, S. 21.

recht gut «belohnt» wird. Es ist in einem gewissen Sinne widersprüchlich, im Urheberrecht sorgfältig den ungeschützten Inhalt von der geschützten Form zu trennen, wenn sich der Schutz im Ergebnis als ein «Alles oder Nichts» auswirken kann. Denn wer den Inhalt auswerten will, braucht eine mit den Mitteln des Urheberrechts kontrollierbare Kopie des ganzen Werkes. Mit KOLLE¹⁴² ist folgende Auffassung zu teilen: «Das Argument, dass die meisten Programme sehr schnell veralten und daher ein Schutzrecht lange vor Ablauf der Schutzfrist gegenstandslos werde, ist in der Allgemeinheit nicht überzeugend.» Diesem Argument zufolge würden bedeutsame rechtliche Auswirkungen den unkalkulierbaren Entwicklungen im ausserrechtlichen Bereich anheimgestellt.

Unrichtig ist es aber wiederum, derartige Überlegungen direkt in den ohnehin belasteten Werkbegriff einfließen zu lassen. Etwas wirklichkeitsfremd ist das zweite Argument im eingangs erwähnten Zitat. Wegen der beschränkten Bearbeitungsfähigkeit des Programmes und der im Verhältnis zur gesamten Schutzfrist kurzen Nutzungsdauer ist kaum anzunehmen, dass noch Mittel aufgewendet werden, um eine überlange Schutzdauer noch weiter zu erstrecken.

IV. Die Neuheit

Die Neuheit¹⁴³ ist nach herrschender, in der Rechtsprechung aber noch nie eindeutig bestätigter Auffassung kein Kriterium für die Schutzwürdigkeit¹⁴⁴ eines Werkes. Es genügen individuelle Züge, auch wenn sie nicht absolut oder innerhalb eines Kulturkreises neu sind. Ebenfalls braucht der dargestellte Gegenstand nicht neu zu sein. Die ausreichende subjektive Neuheit kann, muss aber nicht ein Indiz für die Schutzwürdigkeit sein. Ausser Betracht fällt etwas Neues namentlich dann, wenn es nicht Ausdruck individueller, sondern allgemein-menschlicher Fähigkeiten ist.

Die Koexistenz zweier unabhängig voneinander geschaffener Werke ist möglich. Jeder Urheber erwirbt daran ein selbständiges Urheberrecht¹⁴⁵.

V. Formalitäten

Die Software ist teilweise sehr kurzlebig und während der Entwicklung und der ganzen Dauer ihres Gebrauches zahlreichen Änderungen un-

142 KOLLE G., GRUR 1973, S. 619.

143 Vgl. zur neueren Lehre PEDRAZZINI M., Gedanken zur Neuheit, Schriftenreihe der UFITA, Heft 29, Ehrengabe an EUGEN ULMER, 1965, S. 230ff.

144 Zusammenstellung GIRTH P., S. 34f.

145 HUBMANN H., 4. Aufl. S. 34.

terworfen. Auch diese Gründe sprechen für das urheberrechtliche System, das den Schutz von keinen Formalitäten wie Hinterlegung, Registrierung, Vermerke usw. abhängig macht¹⁴⁶. Der Schutz setzt unmittelbar bei der Entstehung des Werkes ein. Es entsteht kein den praktischen Bedürfnissen stark entgegenstehender Zeitverlust bis zum Wirksamwerden des Schutzes wie bei anderen Schutzmethoden. Der Schutz belastet auch nicht mit anderweitigen administrativen Tätigkeiten und hat daher insgesamt den grossen Vorteil, dass er keine nennenswerten Kosten verursacht.

Ergebnis

Sowohl die Frage des kulturellen Gehaltes wie auch der Umstand, dass der Urberschutz bei der Software nur bei Nebensächlichem einsetzen kann, lassen gefühlsmässig Zweifel aufkommen an der Schutzfähigkeit dieser geistigen Erzeugnisse. Die Bedenken sind weder hinsichtlich des Bestandes dieses Schutzes noch in bezug darauf zu teilen, dass der geringe Schutzzumfang nicht zu genügen vermag, um in Ergänzung und unter Berücksichtigung der anderen tatsächlichen und rechtlichen Schutzmöglichkeiten einen wirksamen Schutz sicherzustellen. Wesentlich ist, dass der krasseste und wahrscheinlichste Verletzungsfall, die unmittelbare und unverändernd nachschaffende Übernahme der Form dieses Leistungsergebnisses, vom urheberrechtlichen Schutz erfasst werden kann. Dieses Hindernis bewirkt bereits in den meisten Fällen, dass der Weg zum Schutzrechtsinhaber vorteilhafter ist. Weiter gewährleistet der im wesentlichen auf die Form beschränkte Schutz auch eine gewisse inhaltliche Exklusivität, weil der Urberschutz es ermöglicht, die Verbreitung des Informationsträgers zu kontrollieren.

Praktikabilitäts Gesichtspunkte im Zusammenhang mit der gerichtlichen Durchsetzung sind nach unserer Auffassung weitgehend systemneutral und können nicht besonders gegen die Tauglichkeit des urheberrechtlichen Systems sprechen. In dieser Hinsicht wird die Wirkung und Bedeutung aller rechtlichen Schutzsysteme im Vergleich zu den tatsächlichen Schutztechniken nur sehr beschränkt sein können, ihnen kann in den meisten Fällen nur eine sehr subsidiäre Funktion zufallen. Das rechtfertigt in einem hohen Masse, von aufwendigen und perfektionistischen Sonderregelungen abzusehen, für deren Wirksamkeit erst noch ein internationales Vertragssystem zustandegebracht werden müsste¹⁴⁷.

146 Das gilt nicht notwendig für Staaten, die nur dem Welturheberrechtsabkommen angeschlossen sind.

147 Siehe auch Rapport der Schweizergruppe der AIPPI 1975/II, Schr. Manusk., S. 7ff, 1974/I, S. 29, S. 37ff, und Hinweise bei KOLLE G., DdA 1977, S. 78.

Das Urheberrecht bietet ein rechtspolitisch verantwortbares Maximum¹⁴⁸ an Schutz bei einem Minimum an insbesondere die zahlreichen kleineren Softwareunternehmen belastenden Formalitäten. Mit einigen geringen allgemeinen Modifikationen oder Klarstellungen im vorhandenen Urheberrecht lassen sich allfällige Inkompatibilitäten beseitigen¹⁴⁹.

148 Oben Kap. 3 C II.

149 Unten Kap. 7.

7. Kapitel: Sonderschutz für Computersoftware de lege ferenda

A. Einleitung

In den letzten Jahren wurden verschiedene Sonderschutzsysteme vorgeschlagen¹. Besonders diskutiert wurden die Studien der WIPO, die Mustervorschriften für den Schutz der Computersoftware entwickelt hat². Den WIPO-Mustervorschriften, die neuesten und bisher auf breiter Basis entwickelten Vorschläge, gelten auch die nachfolgenden Betrachtungen.

Diese Mustervorschriften wollen nicht in dem Sinne verstanden werden, als erforderten sie notwendigerweise die Aufnahme in ein eigenständiges Gesetz zum Schutze der Computersoftware³. In vielen Ländern, so der einleitende Bericht⁴, werden die in den Mustervorschriften enthaltenen Grundsätze möglicherweise lediglich auf eine Klarstellung oder Erweiterung bestehender Rechtsnormen hinauslaufen und könnten – soweit sie nicht schon ohnehin anerkannt sind – in bestehenden Gesetzen Eingang finden, etwa in das Urheberrechtsgesetz, in Rechtsvorschriften über Geschäftsgeheimnisse oder den unlauteren Wettbewerb. Sie versuchen lediglich, den Regelungsgegenstand so vollständig wie möglich zu erfassen und den mit dem Schutz der Software verbundenen Problemen Bestimmtheit zu verleihen.

Sie beruhen im wesentlichen auf einem urheberrechtlichen Lösungsweg⁵. Ihre Grundkonzeption ist jedoch offen und leider teilweise sehr undeutlich.

Eine auf den Mustervorschriften beruhende Gesetzgebung liesse sich ohne weiteres auch zu einer patentrechtlichen bzw. patentähnlichen Lösung ausbauen⁶. Von der Erörterung einer patentähnlichen Konzeption soll

1 Eine wertvolle vergleichende Studie betreffend die Vorschläge von OEHLISCHLEGEL, SENHENN, GALBI, KIRBY, MITI, WIPO u.a. mit Literaturhinweisen ist KINDERMANN M., IIC, Vol 7 (1976), S. 301ff; ferner KOLLE G., DdA 1977, S. 74ff.

2 Mustervorschriften samt Einführung und Kommentierung in WIPO-Publikation Nr. 814, Genf 1978 (englische Originalfassung), (nachfolgend zitiert WIPO-Publ. 814 oder MV); ferner in Ind. Prop. 1977, 259 sowie Copyright 1978, 6; in französischer Sprache in WIPO-Publ. 814, Genf 1978, DdA 1978, 7 und Prop. ind. 1977, 271; in deutscher Sprache Einführung und Mustervorschriften, jedoch ohne einzelne Kommentare in GRUR Int. 1978, S. 286ff.

3 WIPO-Publ. 814, S. 8.

4 WIPO-Publ. 814, S. 8.

5 WIPO-Publ. 814, S. 6: «The model provisions are essentially based on a copyright law approach.» Eingehend auch Seite 15.

6 WIPO-Publ. 814, S. 7, Ziff. 15. Vgl. auch Ausführungen über die in Betracht zu ziehenden Formerfordernisse, Hinterlegungssysteme usw. am gleichen Ort.

an dieser Stelle mit Rücksicht auf den Themenkreis dieser Arbeit Abstand genommen werden. Nachstehend sind bloss die wichtigsten Gründe für die Ablehnung dieser Konzeption zusammenzufassen. Neben den oben⁷ dargelegten rechtspolitischen und von verschiedener Seite bereits im Zusammenhang mit einem Patentschutz *de lege lata* vorgebrachten Bedenken bezüglich Effektivität und Praktikabilität, muss davon ausgegangen werden, dass es keine überzeugenden qualitativen Kriterien gibt, auch andere, bisher mit guten Gründen freigehaltene, nicht-technische Leistungen oder sogenannte Anweisungen an den menschlichen Geist mit einer gewissen Erfindungshöhe, die nicht in eine softwaremässige Lösung eingekleidet sind, von einem so weitgehenden Schutz auszuschliessen. Wenn der wesentliche Algorithmus für eine buchhalterische Lösung schutzfähig sein soll, dann muss es erst recht auch die Lösungsgrundlage, die schutzunfähige Buchhaltungstechnik und z. B. das weite Gebiet der kaufmännisch-organisatorischen Technik, sein.

Im Ergebnis würde man bei einem patentähnlichen Softwareschutz die in den wesentlichen algorithmischen Bereichen nur in beschränkter Zahl verfügbaren interessantesten Anweisungsmuster monopolisieren, was mit Blick auf die Interessen der Allgemeinheit Konsequenzen mit sich bringen müsste, die angesichts der stark zunehmenden Bedeutung des Computers in immer mehr Lebensbereichen nicht zu überblicken wären⁸. Ferner müsste ein patentähnlicher Softwareschutz sehr lückenhaft sein, weil er nur in Fällen mit ausreichender Erfindungshöhe⁹ wirksam wäre. In allen anderen Fällen blieben die erheblichen Aufwendungen ungeschützt. Die wichtigste Verletzungsform, die blosser Übernahme der konkreten Softwarebestandteile, könnte *de lege ferenda* in den weit überwiegenden Fällen gar nicht erfasst werden. Einem Schutzsystem, das eine Breitenwirkung mit geringem Schutzzumfang aufweist, ist gegenüber einem System mit grossem Schutzzumfang für sehr wenige Fälle der Vorzug zu geben.

7 Oben u.a. Kap. 3 C II. Weitere Überlegungen in WIPO-Publ. 814, S. 4 und 5 sowie u.a. KOLLE G., GRUR 1973, S. 619 und dortige Zitate: Die Gefahr ist gross, dass sich derartige Schutzrechte über das wesensmässig mit dem Patentrecht verbundene Mass hinaus hindernd auswirken, weil einzelne Erfindungen im Gegensatz zu Softwareentwicklungen nur sehr beschränkte Bezirke technischen Wirkens erfassen können und ganze Anwendungsgebiete der EDV oder die computerisierte Bearbeitung ganzer Problemkomplexe blockieren bzw. für den Patentinhaber mangels Ausweichmöglichkeiten monopolisieren.

8 Oben Kap. 3 C II. Die Motive für die Freihaltung von konventionellen diagnostischen Methoden der Medizin können sich z.B. kaum unterscheiden von jenen für die Offenhaltung von computerisierten diagnostischen Verfahren.

9 Nach WIPO-Publ. 814, S. 5 könnten vielleicht nur etwa ein Prozent der Programme den Erfordernissen des Patentrechtes genügen. Ähnlich SCHRÖDER O., Referat vom 13. 4. 73 in Rüscliikon (Protokoll), S. 6.

Die nachfolgenden Betrachtungen konzentrieren sich demnach nur auf die urheberrechtsverwandten Lösungsansätze und Ausgestaltungsmöglichkeiten der Mustervorschriften. Sie bezwecken vor allem, anhand der Gegenüberstellung¹⁰ mit dem grundsätzlich tauglichen Urheberschutz *de lege lata*¹¹ Möglichkeiten aufzuzeigen, wie allenfalls vorhandene Grenzen des bestehenden Schutzsystems durch bescheidenere gesetzliche Modifikationen überwunden werden können.

B. Gegenüberstellung der urheberrechtlichen Ergebnisse mit den WIPO-Mustervorschriften¹²

I. § 1 MV: Begriffsbestimmungen

§ 1 Abs. i bis iv MV definiert die Schutzobjekte. § 1 Abs. v MV wird nachfolgend unter § 2 (1) MV behandelt.

Mit dem Begriff «Computersoftware» werden drei Kategorien von Schutzgegenständen erfasst. Es sind dies Computerprogramme, die für jede Art von Maschinen mit informationsverarbeitenden Fähigkeiten geeignet sind, Programmbeschreibungen sowie Begleitmaterial mit beschreibendem oder erläuterndem Inhalt. Nach den Ergebnissen des vorangehenden Kapitels können alle diese Softwareelemente vom Urheberschutz erfasst werden. Die Ergebnisse dieser Arbeit unterliegen der Einschränkung, dass die Untersuchungen nur auf die Verhältnisse des bedeutendsten Gebietes der Informationsverarbeitung, der Betriebsinformatik im weitesten Sinne, und nur auf die wichtigsten Programmarten in diesem Bereich ausgerichtet wurden. Wegen des begrenzten Erfahrungsbereiches des Verfassers bleiben Unsicherheiten, inwieweit vom Urheberschutz auch Programme von informationsverarbeitenden Maschinen erfasst sind, die üblicherweise nicht als Computer bezeichnet werden. Das sind z.B. Programme für mikroprozessorgesteuerte Systeme, die für die Ausführung von speziellen mechanischen Vorrichtungen eingerichtet sind. Es kann angenommen werden, dass bei umfangreicheren Programmen für solche Systeme ebenfalls Spielräume für individuelle Entscheidungen bestehen. Die Mustervorschriften bieten zweifelsohne den Vorteil, dass sie aufgrund der sehr weiten Begriffsbestimmungen ohne eingehendere Untersuchungen unvermittelt und klar auf den Schutzge-

10 Die Gegenüberstellung folgt dem Aufbau der Mustervorschriften Section (§) 1–10.

11 Vgl. Ergebnisse von Kap. 6.

12 Berücksichtigt werden nur die Grundzüge und Ergebnisse des vorgeschlagenen Systems. Es handelt sich nicht um eine umfassende Kommentierung.

genstand hinweisen¹³. Für das Urheberrecht sind die Programme Randscheinungen, die mit Mühe in den Rahmen der üblicherweise beispielhaft aufgezählten Werkkategorien gestellt werden können. Zur Verdeutlichung könnte allerdings auch im Gesetzestext eines Urheberrechtes – selbstverständlich mit den üblichen Vorbehalten – auf Softwarewerke hingewiesen werden¹⁴.

Das Urheberrecht schützt alle Kategorien der in den MV aufgeführten Schutzgegenstände nach einem gleichen Kriterium (Individualität) in einem gleichen Umfang bei allen Verletzungshandlungen. Die MV differenzieren bei den einzelnen Schutzgegenständen bezüglich Schutzzumfang und Verletzungshandlung¹⁵. Im weiteren überlagern die MV in § 3 (Originality) ein allgemeines, nicht einfach zu interpretierendes, urheberrechtsverwandtes, jedoch vermutlich sehr geringes qualitatives Erfordernis für die Erreichung der Schutzzoraussetzungen. Nach der Auffassung des Verfassers können die MV trotz des sehr sachbezogenen und deshalb wohl auch konvenierenden begrifflichen Instrumentariums infolge der eben genannten Differenzierungen erhebliche Probleme bei der praktischen Anwendung des Systems nicht vermeiden. Ebenso wie im Urheberrecht lassen sich bei der Prüfung der materiellen Schutzzoraussetzungen und auch bei der gerichtlichen Durchsetzung¹⁶ schwierige technische Fragen nicht umgehen. Nicht unproblematisch ist die in § 1 MV gewählte Spezifizierung und abschliessende Aufzählung der Schutzgegenstände, d.h. die allgemeine Abgrenzung des sonderrechtlichen Zuständigkeitsbereiches. In der EDV existieren, vor allem im Bereich der immer häufiger werdenden Generatorsysteme, analytische Unterlagen, die in vielen Fällen in den vorgelagerten Bereichen der allgemeinen organisatorischen, administrativen oder kommerziellen Technik anzusiedeln sind und sich deshalb nicht zweifelsfrei den drei Softwarekategorien oder dem allgemeinen Softwarebegriff unterordnen lassen. Die Ge-

13 Das heisst jedoch mit Blick auf die unklaren qualitativen Erfordernisse des § 3 MV noch nicht unbedingt, dass jeder subsumierbare Gegenstand auch wirklich einen Schutz genießt.

14 Vgl. Vorschläge bei WALTER M., GRUR Int. 1974, S. 431. In den USA entspricht dies im wesentlichen auch der Position der CBEMA (Computer and Business Equipment Manufacturers Ass.; deren Mitglieder sind vorwiegend «active creators and manufacturers in the informational and computer arts») zur US-Copyright Revision BILL, H.R. 2223, am 28. 1. 1975 abgegeben an das Committee on the Judiciary The House of Representatives. Die Vorstellungen der an einem wirksamen Schutz interessierten Vereinigung zielen «lediglich» auf eine «clarification of the (Copyright) Bill to provide specially for the adequate copyright protection of data bases, computer programs, and ...». Neben Klarstellungen über die erfassten Verletzungshandlungen (Input in Computer, Vervielfältigung mit Hilfe des Computers usw.) sollte materiell im wesentlichen nur der strafrechtliche Schutz verbessert werden «to the level provided in the Bill for the infringement of sound recordings».

15 Dazu im einzelnen § 5 der MV.

16 Vgl. oben Kap. 6 im einzelnen.

fahr ist z.B. nicht von der Hand zu weisen, dass der Sonderschutz sich auch für den Schutz von Geschäftsformularen¹⁷, Datenbelegen usw. anbieten müsste, was bereits heute eine Domäne des Urheberrechtes ist. Die in den MV angebotenen Begriffsbestimmungen zum Schutzgegenstand schaffen keine allgemeine und überzeugende Trennlinie, um zahlreiche Duplizitäten zwischen Urheberrecht und Sonderschutz zu vermeiden. Im Hinblick auf die teilweise erheblich abweichenden Rechtsfolgen (z.B. bei der Schutzdauer) sollten Doppelspurigkeiten auch nicht so leicht in Kauf genommen werden, wie das die MV in § 9 tun. Aber auch wenn man auf sicherem Boden des Sonderschutzes steht, sind weitere interne, ebenfalls technisch anspruchsvolle Abgrenzungsprobleme vorhanden, weil der Schutz in § 5 MV für die verschiedenen Softwarekategorien differenziert wird. Wie soll beispielsweise zu beantworten sein, bei welchen Steuerangaben von Generatorsystemen (Entscheidungstabellen, Editing-Parameters usw.), die in der Analysephase unter Auslassung einer eigentlichen Programmierungsphase erstellt werden¹⁸, es sich um Programmbeschreibungen, Begleitmaterialien oder Programme handelt?

Von der grundsätzlich einfacheren, wegen den vielfältigen Erscheinungsformen von Softwarewerken wohl einzelfallgerechteren, umfassenderen und bereits etablierten urheberrechtlichen Konzeption sollte nicht ohne Not in Richtung eines Sonderschutzes abgewichen werden, weil auch der grundsätzlich sachbezogener Sonderschutz den Rechtsuchenden nicht von schwierigen technischen Fragen und Unsicherheiten bei notwendigen oder teilweise vermeintlich erforderlichen Abgrenzungen entlasten kann.

Wie SCHWAIGER/KOCKLER¹⁹ richtig feststellen, kommt der Urheberrechtsschutz nur für einen Teil der Leistungen aus dem Gesamtbereich der Datenverarbeitung in Betracht. Dasselbe trifft aber auch für die MV zu, weil sie in urheberrechtsähnlicher Weise den Schutz an den wichtigsten Erscheinungsformen der Software anknüpfen und nicht schlechthin für Leistungen auf dem Gebiet der elektronischen Informationsverarbeitung einen Schutz in Betracht ziehen. Zentrale Bestandteile der auf dem EDV-Markt erheblichen Gesamtleistung, wie Projektmanagement und andere organisatorische, unternehmerische oder technische Leistungen, genießen hier wie dort zu Recht keinen rechtlichen Schutz. In dieser Beziehung sind beide Schutzsysteme ebenbürtig.

17 Vgl. z.B. GRUR-Ber. 1975, Nr. 1462/75: Urheberrechtlicher Schutz eines Computer-Fragebogens, Entscheid LG München I vom 5. 12. 72.

18 Wie oben in Kap. 2 E III 3 und 4 erwähnt, wird in einzelnen Bereichen der Datenverarbeitung bereits in der Analysephase die Ausdrucksweise so kanalisiert, dass Unterschiede zwischen den Softwarekategorien verwischt werden.

19 SCHWAIGER/KOCKLER, S. 361ff.

Als Rechtsinhaber werden gemäss § 1 Abs. v MV diejenigen natürlichen oder juristischen Personen bezeichnet, denen die durch die MV gewährten Rechte gemäss § 2 (1) und § 2 (2) zustehen.

Analog dem fest verankerten Urheberschaftsprinzip, nach dem die Rechte an Werken originär denjenigen Personen zustehen, die sie schufen, gehören nach § 2 (1) MV grundsätzlich auch die Rechte an der Software den Schaffenden²⁰. Die Übertragung der Rechte an andere Personen, insbesondere die Arbeit- und Auftraggeber, kann nach den MV nur aufgrund einer entsprechenden vertraglichen Grundlage erfolgen (§ 2 (1) MV 2. Teil und § 2 (2) 1. Teil). Erleichtert wird die Rechtsübertragung dadurch, dass in § 2 (1) MV, 2. Satzteil, eine Vermutung für den Rechtsübergang aufgestellt wird, wenn die Herstellung der Software zu den im Arbeitsverhältnis vorgesehenen Pflichten des Arbeitnehmers gehört und keine abweichende Vereinbarung vorliegt. Interessant ist in rechtsvergleichender Sicht die Lösung im § 201 des neuen US-Copyright Act (1976), die anfänglich das Recht dem Urheber zuspricht, jedoch eine vollständige Übertragung vorsieht und im Gegensatz zum schweizerischen und deutschen Recht eine Vermutung aufstellt, dass das Urheberrecht an Arbeitnehmer- oder Auftragswerken dem Arbeit- bzw. Auftraggeber zusteht, sofern nicht etwas anderes ausdrücklich vereinbart wurde²¹.

Die MV lehnen sich eng an übliche Regelungen in den Urhebergesetzen an, ohne die besonderen Probleme dieses Schutzgegenstandes zu beachten²². Bei den einzelnen Softwareelementen liegt meistens eine Akkumulation von eng verzahnten Schaffensbeiträgen einer unbestimmten Personmehrheit vor. Oft lassen sich die einzelnen Beiträge wegen ihres überwiegend sachlich bestimmten unpersönlichen²³ Charakters nicht einmal voneinander abgrenzen oder bestimmten Schaffenden zuordnen. Es lässt sich im einzelnen nicht immer zuverlässig feststellen, von wem welche Rechte abzuleiten sind. Andere Lösungsansätze, die im Zusammenhang mit einem Urheberrecht *de lege ferenda* diskutiert werden²⁴, scheinen uns wesentlich geeigneter zu sein.

20 WIPO-Publ. 814, S. 13: «In principle, the rights under the Law belong to the creator of the computer software».

21 Vgl. Beitrag BRAVEMANN P. E., Das neue US-Copyright-Law, Schriftenreihe der UFITA, H. 61, 1978, S. 77ff.

22 Oben Kap. 6 F II.

23 Unpersönlich im Sinne, dass keine Rückschlüsse auf die Person des Schaffenden möglich sind.

24 Vgl. die von LARESE W. aufgeführte Literatur unter Kap. 6 F II und nachfolgende Zitate.

Allenfalls müsste entsprechend den wirtschaftlichen Interessenlagen wie bei ähnlichen Leistungen eher am Aufwand von organisatorischen und finanziellen Mitteln angeknüpft werden. Es überrascht angesichts der Kritik, die im Zusammenhang mit der Software am bestehenden urheberrechtlichen Zuordnungssystem angebracht wurde²⁵, dass die MV nicht deutlich von einer urheberrechtlichen Lösung abweichen. Auf jeden Fall ist dieses System kein wesentliches Hindernis für die Gewährung eines Urheber-schutzes für Software. Es erfordert aber in den meisten Fällen ein bewegliches vertragliches Sekundärsystem, das eine sinnvolle einheitliche Verwertung des Schutzgegenstandes sicherstellt und dafür die erforderlichen Rechte in einer Hand vereinigt. Die Probleme ähneln wiederum jenen beim Filmwerk. In vergleichbarer Weise sollte das Gesetz durch Vermutungsregeln erleichtern, dass der Softwareunternehmer von den Arbeitnehmern²⁶, Beauftragten und sonstigen Urhebern die für die Verwertung unerlässlichen Rechte erhält. Notwendig für eine sinnvolle Verwertung sind die ausschliesslichen oder nicht ausschliesslichen Rechte für die Bekanntmachung, die Vervielfältigung und die Bearbeitung oder Umgestaltung in beliebigem Umfange. Bei der Software erfordert der Nutzungszweck in der Regel, dass beliebige Änderungen im Zusammenhang mit Fehlerbehebungen, Unterhaltsarbeiten und Anpassungen an neue Anwendungs- oder Hardwareerfordernisse vom jeweiligen Rechtsinhaber vorgenommen werden können.

Für die in Erfüllung eines Vertrages geschaffenen Werke findet sich ein geeigneter Lösungsansatz in VE II Art. 27 URG. Wünschenswert sind ähnliche Sonderregelungen, wie sie in VE II Art. 49 und 50 URG für Film- und Fernsehwerke geschaffen wurden. Dem Softwareunternehmer sollten unter dem Vorbehalt abweichender Vereinbarungen vermutungsweise die für die Verwertung des Gesamtwerkes erwähnten notwendigen Verwendungsarten zugesprochen werden. Mit Rücksicht auf den Charakter der Software und den Umstand, dass sie meist stark fremdbestimmt im Rahmen eines Arbeits- oder Auftragsverhältnisses hergestellt wird, rechtfertigt es sich

25 Oben Kap. 6 F II.

26 Eingehender über die dogmatische Möglichkeit der Entstehung von Urheberrechten beim Arbeitgeber mit Vorschlag zur analogen Regelung wie im Arbeitnehmererfindungsgesetz REHBINDER M., Recht am Arbeitsergebnis und Urheberrecht, UFITA Bd. 66 (1973), S. 125f und 148; ders. ZSR 91 I (1972), S. 1ff. Ferner siehe auch RECHER M., Der Arbeitnehmer als Urheber und das Recht des Arbeitgebers am urheberschutzfähigen Arbeitsergebnis, Diss. Zürich 1975. Eine nähere Regelung dieses Fragenkreises ist in Aussicht gestellt. Nach REHBINDER M., Schweizerisches Arbeitsrecht, 5. Aufl. 1979, S. 90, ist bei den urheberrechtlich geschützten Werken die (geltende) Rechtslage nach überwiegender Auffassung entsprechend der Regelung für die gewerblichen Muster und Modelle (Art. 332a OR) zu beurteilen, d.h. für die in Ausübung der dienstlichen Tätigkeit und in Erfüllung der Dienstpflicht geschaffenen Werke gehen die Nutzungsrechte ex lege auf den Arbeitgeber über, soweit der Zweck des Arbeitsverhältnisses dies erfordert.

nicht, de lege ferenda zum Schutze des Urhebers vom Prinzip der Vertragsfreiheit dadurch abzuweichen²⁷, dass eine sehr weitgehende Übertragung der Verwendungsrechte durch zwingende Normen vereitelt oder erschwert wird. Von sehr begrenzter praktischer Bedeutung ist angesichts des Charakters der Softwareschöpfungen, dass bei einer Rechtsübertragung ein gewisser Kernbereich der Persönlichkeitsrechte nach einem Teil der Lehre unberührt bleiben muss, soweit nicht in zulässigem Rahmen bereits im voraus ein Verzicht auf die Geltendmachung von Ansprüchen aus deren Verletzung ausgesprochen wird.

III. § 3 MV Originalität

In Anlehnung an ein Kriterium, das mit gleicher Bezeichnung in den Urhebergesetzen der meisten Länder enthalten ist, fordern die MV für die Schutzfähigkeit die Originalität. Die Software braucht aber trotz der terminologischen Übereinstimmung nicht im Sinne des Urheberrechtes originell zu sein. Die MV erklären, sehr unverbindlich und jedem Land Freiheiten in der Interpretation lassend, dass die Software lediglich «the result of its creator's own (independent) intellectual effort»²⁸ sein muss. Die Worte «intellectual effort» und die dazugehörigen Erläuterungen²⁹ legen die Interpretation nahe, dass die Qualität der in der Software zum Ausdruck kommenden geistigen Leistungen irgendwo oberhalb des Stereotypen, Naheliegenden sein muss und eine Ebene erreichen kann, die mit der verlangten Erfindungshöhe im Patentrecht zu vergleichen ist. Ein Softwareergebnis wird desto eher den qualitativen Anforderungen dieses Schutzsystems entsprechen, je grösser der quantitative und qualitative geistige Einsatz ist. Quantitative und qualitative Momente können kompensierend wirken³⁰. Schutzfähig ist bereits eine «kleine Softwareerfindung», wobei der konzeptionelle Gehalt schutzbegründend, weil auf den geistigen Einsatz hinweisend, aber nicht schutzfähig ist (vgl. § 4 MV). Geschützt sein sollen nach diesen MV Softwareschöpfungen, die unabhängig geschaffen wurden und mindestens einen so grossen Umfang aufweisen, dass auch bei routinemässiger, an gängige Lösungsmuster anlehnender Entwicklungsarbeit eine Schwelle überschritten wird, bei der sich die Unabhängigkeit der eigenen geistigen

27 Über aktuelle Tendenzen siehe VE II URG, Erläuterungen Expertenkommission, S. 19ff.

28 WIPO-Publ. 814, S. 15. Dem erklärten Ziel der WIPO, durch die MV zu einer wünschbaren Vereinheitlichung des Softwareschutzes beizutragen, ist es nicht gerade förderlich, wenn ein neues Qualitätskriterium postuliert wird, das ebenfalls grosse Interpretationsfreiheiten zulässt. Nähere Erläuterungen auch bei KINDERMANN IIC No. 3 1976, S. 323.

29 WIPO-Publ. 814, S. 15.

30 WIPO-Publ. 814, Folgerung aus Beispielen auf S. 15.

Leistung überhaupt ausweisen kann. Ein vergleichbares generalisierendes Kriterium, nach dem weitgehend auf den Umfang der Leistung abgestellt werden kann, wurde auch bei der urheberrechtlichen Würdigung postuliert³¹. Beim Urheberschutz ist neben der Formgebung auch die übrige geistige Leistung in Betracht zu ziehen, die selbst bei der Verarbeitung von zahlreichen naheliegenden Lösungselementen bei genügendem Umfang nach den allgemeinen Wahrscheinlichkeitserwartungen in Kombination und Anordnung in eine Individualität hineinwachsen kann. Aufgrund der Ergebnisse der vorangegangenen Betrachtungen, denen zufolge in der Regel bereits in kleineren Programmen mit 500–1000 Zeilen Individualitäten nachgewiesen werden können, besteht kein Grund, von einer Originalität im urheberrechtlichen Sinne Abstand zu nehmen, um auf sonderrechtlichem Wege zu ähnlichen Ergebnissen zu gelangen. Falls dieses urheberrechtliche Erfordernis im Sinne der MV preisgegeben oder verwässert werden sollte, so bestehen überdies keine überzeugenden Gründe mehr, anderen Sorgenkindern des Urheberrechtes wie technischen Plänen, Editio princeps, Kartenwerken, Druckwerken, Bildträgern usw. einen ähnlichen, ebenso einfach zu erreichenden Schutz vorzuenthalten³².

IV. § 4 MV: Konzepte

In dieser Bestimmung kommt die Nähe zum Urheberschutz am deutlichsten zum Ausdruck. Auch die MV wollen nur die Form («form of expression»), in die Softwarekonzepte oder «computerisierte» Verfahren eingekleidet sind, schützen und nicht die Konzepte selber³³. Wie beim Urheberschutz dürfen Konzepte und selbstverständlich Algorithmen von Dritten übernommen werden. Lediglich die Ausdrucksform eines Vorbildes darf nicht kopiert werden. Zum geschützten Bereich gehört auch die Struktur der Software³⁴. Der Schutzzumfang der MV stimmt mit jenem des Urheberrechtes überein.

Die MV gehen offenbar davon aus, dass in der Ausdrucksform genügend Spielräume vorhanden sind. Es wird stillschweigend unterstellt, dass aus deren Schutz auch bei zulässigen konzeptionellen Abhängigkeiten nicht eine Mitteilungsexklusivität für einzelne konzeptionelle Elemente entstehen kann. Ohne Rückwirkung auf die freigehaltenen inhaltlichen Bereiche müs-

31 Oben Kap. 6 E IV 1.

32 Hierbei wären allgemeinere Lösungsansätze vorzuziehen (vgl. etwa RAU G., *Antikunst und Urheberrecht*, Berlin 1978 und PURTSCHERT J., *Der Schutz des unmittelbaren Ergebnisses einer Arbeits- und Unternehmensleistung*, Diss. Freiburg i. Ue. 1974).

33 WIPO-Publ. 814. S. 16.

34 WIPO-Publ. 814. S. 16: «the structure of another person's software», «the arrangement of concepts».

sen bei der Anlehnung an Vorlagen einigermaßen zumutbare Alternativen in der Form vorhanden sein. Zur Erreichung des gleichen Zwecks mit einer identischen konzeptionellen Grundlage sollten ohne wesentliche Änderung der technischen Funktionalität auch andere Formen geschaffen werden können, als sie in der Vorlage vorhanden sind³⁵. Das muss wieder in die Nähe des urheberrechtlichen Ausgangspunktes zurückführen, weil das Problem nur zufriedenstellend gelöst werden kann, wenn eine Darstellungsweise Individualität aufweist. Nur das urheberrechtliche Erfordernis der Individualität gewährleistet, dass die schützbar Ausdrucksform im weitesten Sinn auf Positionen zurückversetzt wird, wo ein Konzept mehreren abhängigen Nachschöpfen ohne wesentliche Behinderungen zur Verfügung steht. Untragbar wäre eine derartige faktische Ausschlusswirkung bereits dann, wenn von der Anwendung von wichtigen Programmiersprachen Abstand genommen werden müsste, um ein gleiches Konzept noch verwirklichen zu können. Zurückhalten ist auch mit Rücksicht auf die Verwendung der EDV-Literatur, die in einer programmsprachlichen Form algorithmische Elemente für die Zwecke der Schulung, Wissenschaft u. dergl. bekanntmacht. Vorsicht ist allgemein am Platze, weil gewisse Programmiersprachen für umfangreiche Lösungselemente gelegentlich über weite Strecken nicht mehrere vertretbare Alternativlösungen anbieten. Die unter Umständen lückenhafte und nur punktuell wirkende urheberrechtliche Konzeption ist deshalb klar zu bevorzugen.

Hinzuweisen ist noch darauf, dass sowohl in § 4 MV wie auch in § 5 Abs. (iv) und (v) MV genauere Hinweise fehlen, inwiefern die Ausdrucksform nicht übernommen werden darf und mit Bezug auf welche Elemente eine Programmbeschreibung und ein davon abgeleitetes Computerprogramm (§ 5 Abs. iv und v MV) ungleich oder im wesentlichen ungleichartig sein müssen. Den MV fehlt ein klares grundlegendes Konzept. Dem abhängigen Nachschöpfer ist es nicht möglich, die verbotenen Bezirke seiner an eine fremde Form anknüpfenden Handlungen einigermaßen klar abzugrenzen.

V. § 5 MV: Rechte des Inhabers

Den in § 1 Abs. i bis iii MV definierten drei Softwarekategorien wird in den MV nicht der gleiche Schutz gewährt. Es wird jedoch darauf verzichtet, die in einer sehr kasuistischen Form geregelten Einzelheiten dieses Artikels eingehender zu besprechen.

35

Vgl. Überlegungen bei PURTSCHERT J., S. 203; auch PEDRAZZINI M., in Festschrift KARL OFTINGER, Zürich, 1969, S. 207ff, S. 221.

1. § 5 Abs. i, ii MV: *Geheimhaltung*

Die beiden Absätze enthalten für alle Softwarebestandteile und deren Verkörperung Abwehrrechte, die aus Vertrag oder aus der Verletzung von Fabrikations- und Geschäftsgeheimnissen³⁶ (Art. 1 Abs. 2 lit. f und g UWG) abgeleitet werden können. In beschränktem Masse kommt auch das Urheberrecht in Betracht, welches den Urhebern das Erstveröffentlichungsrecht (§ 12 Abs. 1 UhrG) bzw. das Recht zur Bekanntmachung (VE II Art. 13 Abs. 1 URG) und das Recht der ersten Inhaltsmitteilung (§ 12 Abs. 2 UhrG, VE II Art. 13 Abs. 1 URG) vorbehält. Letztgenanntes Recht ist bei freien Erkenntnissen nur wirksam, wenn die Erkenntnisse bei der Veröffentlichung nicht von der urheberschutzfähigen Gestaltung gelöst wurden³⁷. Die bestehenden Rechtsbehelfe dürften jedoch ausreichen, um entsprechend der Absicht der MV den Geheimhaltungsinteressen während der Entwicklungsphase gerecht zu werden.

2. § 5 Abs. iii MV: *Vervielfältigung*

Dieser Absatz betrifft die dem Urheber ebenfalls vorbehaltenen Rechte zur körperlichen und unkörperlichen Reproduktion des Werkes³⁸.

3. § 5 Abs. iv und v MV: *Unveränderte nachschaffende Übernahme oder Nachahmung von Programmen und Programmbeschreibungen sowie Herstellung von gleichartigen Programmen nach Programmbeschreibungen*

Wie bereits bei der Besprechung des § 4 MV festgestellt, besteht eine weitgehende Übereinstimmung der Schutzbereiche. Teilweise umfassender, als es die MV vorsehen, kann das Urheberrecht im Bereich der Ausdrucksform im weiteren Sinn bewirken, dass bei Nachahmungen von Softwareergebnissen früherer Phasen, das sind neben Programmbeschreibungen auch Flussdiagramme, Bildschirm- und Druckbilder u. dergl., und bei Programmen auf Alternativen auszuweichen ist. Dasselbe gilt aufgrund der den Urhebern zustehenden Bearbeitungsrechte für das Verhältnis zwischen den Softwareergebnissen verschiedener Entwicklungsphasen. Wie die MV kann auch das Urheberrecht wegen der freien Verfügbarkeit des Konzeptes nur bedingt verhindern, dass der Nachschaffende mit wesentlich geringerem

36 Siehe oben Kap. 4 C.

37 MÖHRING-NICOLINI, Kommentar zum Urheberrechtsgesetz, § 12 Abs. 2 UhrG, Anm. 7 und 8. Oben Kap. 5 B II 5.

38 Näheres oben in Kap. 6 F I.

Aufwand zum gleichen Ergebnis gelangen kann. Beide Schutzsysteme gewährleisten, dass der aufwendige Umweg über die eingehende Detailanalyse und Umgestaltung der Vorlage, die Kodierung oder Dokumentation, Erfassung, Kontrolle und das Testverfahren notwendig ist, was in der Regel eine beträchtliche Schutzwirkung erzeugt³⁹.

4. § 5 Abs. vi MV: Gebrauch und Einspeicherung

Einspeicherungen von Softwarewerken in einen Computer sowie Ausgaben auf Datenzwischenträger sind urheberrechtlich geschützte Werkverwendungen⁴⁰. Das gilt nicht für die Benutzungen der Software, die lediglich den maschinellen Ablauf steuern⁴¹. Der Programmgebrauch ist keine Werkvermittlungsart und steht deshalb ausserhalb der urheberrechtlichen Befugnisse. Diese Lücke scheint unbedeutend zu sein, weil der Gebrauch mit Sicherheit die urheberrechtlich kontrollierbare Verbreitung von Datenzwischenträgern, erstmalige Einspeicherung in den Computer, computerinterne Vervielfältigung sowie bei der Programmwartung usw. die Ausgabe auf Druckern oder Sichtgeräten voraussetzt. Ferner ist der Gebrauch von Programmteilen in vielen Fällen mit der Verbreitung von anderen Werkteilen verbunden. Es bestehen also ausreichende Alternativen für ein Eingreifen des Urheberschutzes.

5. § 5 Abs. vii MV: Verschiedene Verbreitungszwecke

Diese Absätze detaillieren Verfügungsformen, die mittelbar oder unmittelbar aufgrund der Rechtsstellung des Urhebers und den ihm zustehenden Verwendungsbefugnissen kontrolliert werden können.

VI. § 6 MV: Rechtsverletzungen

Das Urheberrecht richtet sich ebenfalls nur gegen ein Nachmachen im subjektiven Sinn. Unabhängige Parallelschöpfungen können auch unter dem Urheberrecht koexistieren. Soweit in unabhängigen Schöpfungen Übereinstimmungen zu beobachten sind, muss die individuelle Eigenschaft in Frage gestellt werden⁴².

§ 6 (3) MV betrifft einen Vorbehalt, der auf Art. 5ter der Pariser Übereinkunft zum Schutz des gewerblichen Eigentums zurückgeht⁴³.

39 Oben Kap. 6 E.

40 Oben Kap. 6 F I, eingehend.

41 Oben Kap. 6 F I.

42 Zur Neuheit vgl. oben Kap. 6 F V.

43 Vgl. WIPO-Publ. 814, S. 20

VII. § 7 MV: Schutzdauer⁴⁴

Die Schutzdauer der Software wird in den MV mit Berücksichtigung ihres gewerblichen Charakters auf 20 bzw. maximal 25 Jahre beschränkt. Die Berechnung der Schutzfristen knüpft teilweise an andere Ausgangspunkte an. Die MV sind zweifellos angemessener. Sie berücksichtigen eine realistische Schutzerwartung, die beim jetzigen Gang der Entwicklung im allgemeinen in 10 bis 20 Jahren nicht mehr vorhanden ist.

Die Unverhältnismässigkeit der urheberrechtlichen Schutzdauer lässt sich noch in vielen verwandten Bereichen des urheberrechtlich geschützten Schaffens beobachten und kann wohl nicht von ausschlaggebender Bedeutung sein, um ein besonderes Schutzsystem ins Auge zu fassen.

VIII. § 8 MV: Rechtsbehelfe

Gleiche Rechtsbehelfe wie die MV stellen auch das Urheberrecht und das UWG bei ähnlichen Verletzungstatbeständen (vgl. § 5 Abs. i und ii MV für das UWG betreffende Verletzungshandlungen) zur Verfügung (Art. 52ff URG, VE II Art. 52ff URG sowie Art. 2 Abs. 1 lit. a ff UWG).

SCHWAIGER/KOCKLER⁴⁵ erachten beim Urheberschutz den wirtschaftlichen Schutzzumfang, vor allem den möglichen Umfang der Schadenersatzansprüche, als völlig unzureichend, weil in keinem Fall der Gesamtwert der Software geltend gemacht werden kann. Eine Urheberverletzung würde nur den sehr sekundären individuellen und nicht den primären Bereich der geistigen Leistung berühren, der das wertvollere gemeinfreie Konzept bzw. den Algorithmus enthält. Die Schadensbemessung könnte dadurch nicht wesentlich über die Kosten einer Änderung des benutzten Programmes in ein aus urheberrechtlicher Sicht völlig unabhängiges Parallelprogramm hinausgehen.

Das ist aber keine besondere «Unzulänglichkeit» des Urheberschutzes. Nach dem Plan der MV ist der gleiche wertvollere Bereich ebenfalls gemeinfrei, deshalb kann die Übernahme des wesentlichen verstandesmässigen Gehaltes ebenfalls nicht oder nur in einem sehr beschränkten Rahmen bei der Schadensberechnung zugrundegelegt werden. Der Kommentar der MV⁴⁶ schweigt zu diesem Problem. Neben der Unterlassung könnte der Kläger immerhin aufgrund von beiden Schutzsystemen wenigstens eine übliche Verwendungsvergütung erwirken.

44 Ausführungen oben Kap. 6 F III.

45 SCHWAIGER/KOCKLER, S. 366.

46 WIPO-Publ., S. 22.

Soweit diese Bestimmungen die kumulative Anwendung der spezialrechtlichen Normen und der Regeln gegen den unlauteren Wettbewerb vorsehen, entspricht dies der herrschenden Lehre⁴⁷. Ein auf der Grundlage der MV eingerichteter Sonderschutz *de lege ferenda* würde in weiten Bereichen zu Parallelitäten mit dem Urheberschutz führen. Der Patentschutz berührt die von einem Urheberschutz oder einem nach den MV konzipierten Sonderschutz erfassten Bereiche nicht und kommt bei der Computersoftware nur in einem sehr beschränkten Rahmen in Frage⁴⁸. Die Bestimmung legt keine Prioritätsordnung zwischen den infragekommenden Schutzsystemen fest und nimmt eine aus verschiedenen Gründen zu vermeidende Koexistenz von Schutzwirkungen aus bestehenden Systemen in Kauf.

Ergebnis

Aus den kurzen Ausführungen dieses Kapitels folgt:

1. Im bestehenden schweizerischen Recht, insbesondere im Bundesgesetz betreffend das Urheberrecht an Werken der Literatur und Kunst und im Bundesgesetz über den unlauteren Wettbewerb (bei letzterem in erster Linie im Bereich des Schutzes von Fabrikations- und Geschäftsgeheimnissen), sind ausreichende Grundlagen vorhanden, um die Computersoftware sehr weitgehend in dem von den WIPO-Mustervorschriften vorgezeichneten Rahmen zu schützen⁴⁹, ohne dass hinsichtlich der Praktikabilität und der Schutzwirkung im Vergleich mit dem vorgeschlagenen Sonderschutzrecht besondere Nachteile oder Ungereimtheiten in Kauf zu nehmen sind.

2. Von der Ausarbeitung eines Schutzsystems *sui generis* kann deshalb abgesehen werden. Ein auf dem Boden des Urheberrechtes stehender, in Einzelfragen sachgerechterer Sonderschutz⁵⁰ würde weitgehend auf Doppelspurigkeiten mit dem vorhandenen Urheberschutz hinauslaufen. Gesichtspunkte der Gleichbehandlung sprechen ferner gegen Normierungen, die sich an Bedürfnissen eines sehr beschränkten Bereiches gewerblichen Schaffens orientieren. Das müsste zu einer im Ansatz bereits heute erkenn-

47 TROLLER A., IG II, S. 1103f mit Hinweisen.

48 Oben Kap. 4 D.

49 Ähnliches Ergebnis für die BRD bei KOLLE G., GRUR Int. 1974, S. 450 (Bericht der deutschen Landesgruppe im Rahmen der Int. Vereinigung für gewerbl. Rechtsschutz (AIPPI); für die Schweiz im Rapport der Schweizergruppe, AIPPI, Frage 57 B, Schreibmaschinenmanuskript vom 3. 4. 75, S. 7.

50 Vgl. begriffliches Instrumentarium (§ 1 und § 4 MV), Gebrauchsschutz (§ 5 Abs. vi MV), Schutzdauer und Berechnung (§ 7 MV).

baren Zersplitterung des Schutzes von geistigen Gütern führen. Im weiteren kann – wie in den vorstehenden Betrachtungen mit wenigen Beispielen angedeutet – auch eine Spezialgesetzgebung die Rechtsuchenden bei der gerichtlichen Durchsetzung von schwierigen technischen Fragen, die die Eigenschaften des Schutzgegenstandes betreffen, nicht entlasten. Deshalb ist zumindest unter dem Gesichtspunkt der Praktikabilität mit Bedenken zurückzuhalten, weil Schutzobjekte in ein primär auf ästhetische Gebiete ausgerichtetes Rechtssystem eingebracht werden, deren Beurteilung in überwiegendem Masse Kenntnisse über zweckgerichtetes gewerbliches Schaffen erfordert.

3. Die Bedeutung und Wirksamkeit des rechtlichen Schutzes wird stark beeinflusst durch die im Schutzgegenstand innewohnenden Eigenschaften und dessen Umgebung: Hinzuweisen ist nochmals auf die grosse Bedeutung des Selbstschutzes und von Massnahmen der Datensicherung⁵¹. Weiter ist zu beachten, dass wegen der Komplexität, des Umfanges und anderen Eigenarten bereits ein sehr lückenhafter Schutz mit geringem Schutzzumfang erhebliche Schutzwirkungen erzeugen kann. Von Bedeutung ist auch, dass der Schutz ohne weitere Formalitäten, ohne zeitliche Verzögerung und ohne nennenswerte Kosten wirksam wird und auch sonst kaum praktische Nachteile hat⁵².

Die Besonderheiten des Schutzgegenstandes und die Probleme bei der Aufdeckung von Rechtsverletzungen und der gerichtlichen Durchsetzung lassen annehmen, dass dem Rechtsschutz nur eine sehr subsidiäre Funktion zukommen kann. Es ist zu vermuten, dass das Bedürfnis nach besonderen gesetzgeberischen Anstrengungen eher gering ist⁵³.

4. Obwohl der urheberrechtliche Schutz noch kontrovers ist, weisen viele Zeichen darauf hin, dass man die Computersoftware im Urheberrecht beheimaten will. Gerichtsentscheide liegen jedoch nicht vor⁵⁴.

Die Möglichkeit einer Anwendung bestehender Gesetze hätte zudem den Vorteil, dass der Schutz aufgrund völkerrechtlicher Vereinbarungen in vielen Ländern wirksam wäre, was angesichts des internationalen Charakters der datenverarbeitenden Industrie ein unbedingtes Erfordernis ist.

51 Oben Kap. 3 B.

52 MOOERS C. N., ACM Comp. Surv., S. 68.

53 In den USA bezeichnete selbst eine der gewichtigsten Interessenvertreterinnen der Softwareindustrie, die CBEMA, in einem Bericht an die CONTU den beschränkten Urheberschutz als die am meisten bevorzugte Schutzform (Hinweis bei MYERS E. D., Datamation 3/78, S. 132 und Zitat oben Kap. 7 B I).

54 Zu einzelnen richterlichen Stellungnahmen vgl. PORTIN G., GRUR Int. 1975, S. 19 (finnischer Prüflisten-Fall); Hinweise ohne nähere Bezeichnung der beiden Fälle bei KOLLE G., DdA 1977, S. 72.

Anhang

Nr. 1 Die wichtigste Software für Datenverarbeitungsanlagen

Systemsoftware

Betriebssysteme

Organisationsprogramme

Auftragsverwaltung
Prozeßverwaltung
Datenverwaltung
Datenfernverarbeitung

Übersetzungsprogramme

Assembler
Kompilierer für:
ALGOL
FORTRAN
COBOL
PL 1
Listprogrammgenerator

Dienstprogramme

Sortier- Misch- Generator
Umsetzprogramme
Testhilfen
Bibliothekswartung

Anwendersoftware

Allgemein

z.B. Netzplanprogramme
Management-Informationssysteme
Operations Research
Informations-Wiedergewinnungssysteme

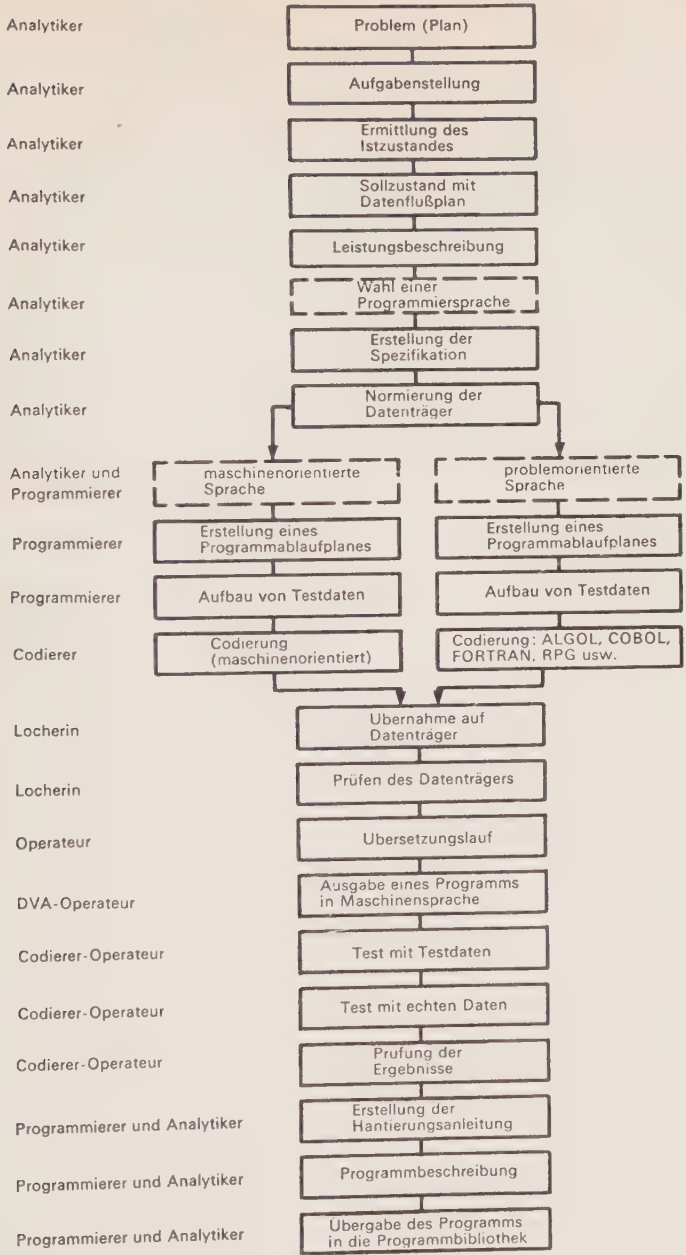
Branchenbezogen

z. B. Lagerbestandsführung
Fertigungssteuerung
Bauabrechnung

Quelle: LÖBEL/MÜLLER/SCHMID, S. 169

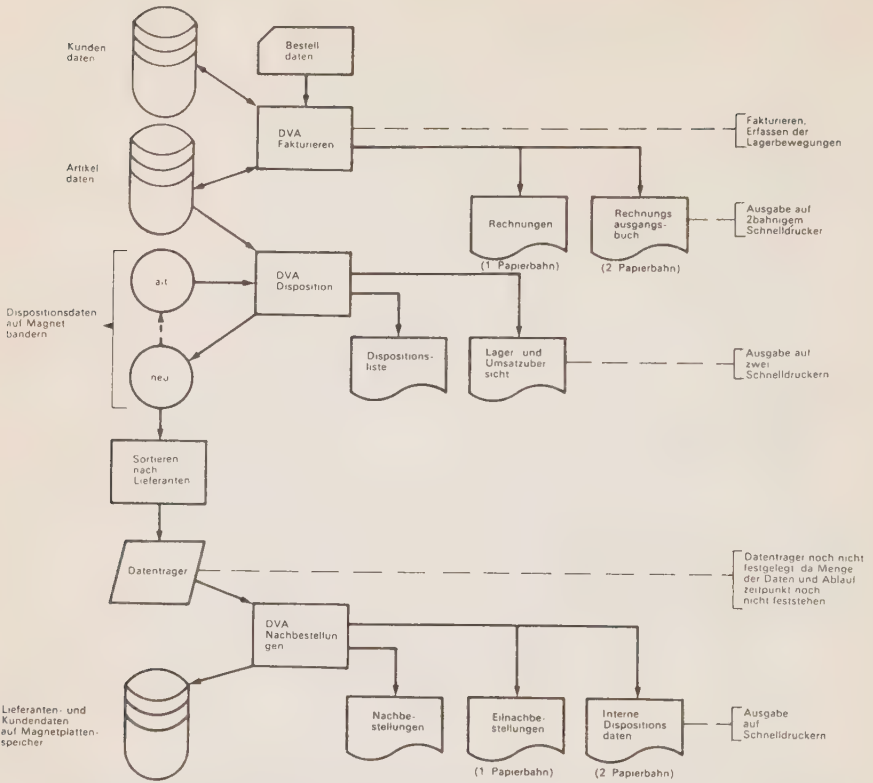
Aus Platzgründen handelt es sich bei den im Anhang gezeigten Beispielen nur um Fragmente, die in dieser Form noch keinen Werkcharakter haben.

Nr. 2 Logischer Ablauf bei der Erstellung eines Programmes



Quelle: LÖBEL/MÜLLER/SCHMID, S. 170

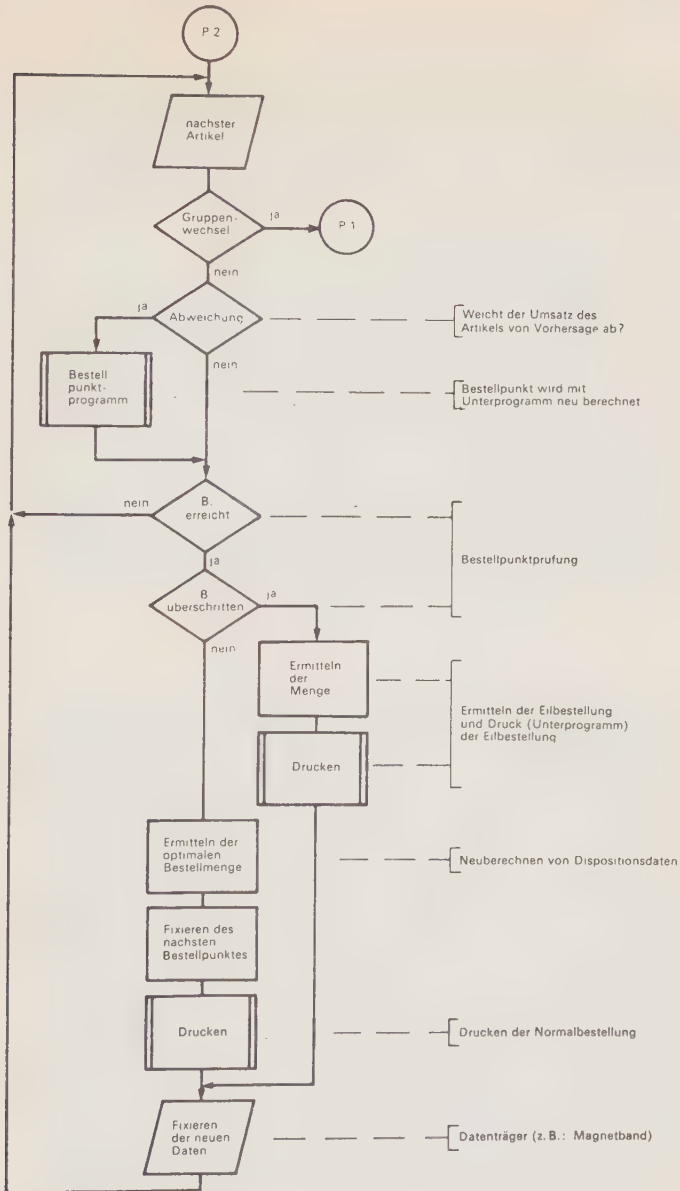
Nr. 3 System-/Datenflussdiagramm



Beispiel eines Datenflußplanes für das Fakturieren auszuliefernder Ware, die Lagerdisposition und das Drucken von Nachbestellungen

Quelle: LÖBEL/MÜLLER/SCHMID, S. 175

Nr. 4 Programmablaufplan



Beispiel für einen Ausschnitt aus einem Programmablaufplan für die Lagerbestandsüberwachung und das Drucken von Nachbestellungen

Quelle: LÖBEL/MÜLLER/SCHMID, S. 176

ORDER ENTRY FIRMA 01 KUNDE 00546001 AUFTRAG 0005003 ACD 0 STATUS 0

61 G. KLEINER
62 EDV-ABTEILUNG
63 BAHNHOFSTRASSE 9
64 5200 BRUGG
65

RECH G. KLEINER
EMPF TV/RADIO
BADENERSTRASSE 17
5200 BRUGG

AUFTOT 0.00
SALDO 860.00
KREDIT 5000.00
DIFF 4140.00
BONITÄT 0

01 STATUS 0
02 AUFTR-DATUM 20/02/79
03 LIEFERSCHEIN 0000000
04 LIEF-DATUM 20/02/79
05 SACHBEARBEITER HK
06 BESTELLART 2
07 AUSLIEFERART 0
08 TOUR 00
09 VERSANDKOSTEN
10 VERPACKUNG

11 RECHNUNGS-EMPF 00546001
12 VERTRETER 0002
13 KUNDENKATEGORIE 00
14 BRANCHE 09
16 ZAHLUNGSKONDITION 01
17 VERFALLDATUM
18 MUSTCODE 3
19 BONUSCODE 0
20 VERSANDTEXT 2. HD. HERRN K. HAUSER

21 PREISCD 4
22 ZEIT 16.10
23 SAMMELFAKT 1
24 AB LAGER 00
26 KLEINHAUF 00
TEL
BAHNK 0049

EINGABEDATEN FELDNUMMER ** FELDINHALT

ORDER ENTRY FIRMA 01 KUNDE 00546001 AUFTRAG 0005003 ACD 0 STATUS 0

61 G. KLEINER
TERMIN

RECH G. KLEINER
RA DISPO-MENGE

AUFTOT 2371.00
VERFUEGBAR

6/03/79 00000001 0199
10/03/79 00886001 0004268
15/03/79 00000001 0218
18/03/79 00546001 0004777

L-BEST 200
K-RES 50-
L-BEST 200
K-RES 30-

320
270
470
440

POSC ARTIKELNUMMER B-NG C-NG BRUTPREIS RAB 1 RAB 2 NETPREIS M.D. TERMIN 00

0100 4501 150
BEZ1 JOURNALROLLEN IPP2 14/2L LAME FREI 120 LMG 120

ENDE DISPO ****

Beispiel für Bildschirmbilder aus einem Programm für die kommerzielle Auftragsbearbeitung.
Quelle: NCR (Schweiz), COPAS, Systemdokumentation

Nr. 6 Druckbilder/Drucklayout

NCR IPPS IW018--00

DATUM 79/02/28 ZEIT 17/36/20

BESTELLVORSCHLAG FUER EINKAUFSSTELLE 1 VON GRUND 101 VOM 28/02/79

ARTIKEL

2001

FARBAND 644 MATRIX
AD20 FARBAND

1 LIEFER TERMIN JJTTT

.....

3 LIEFERANT / BEST.NR

..... /

4 1 = OFF. / 2 = BESTELLT

..

6 BESTELLTE MENGE

..... SK

LAGER-BESTAND 168.00 SK

12 INTERNER AUFTRAG

.....

BESTELLPUNKT 160 SK

13 BESTELL TERMIN JJTTT

.....

SICHERN.BEST. 60 SK

15 BESTELLER

.....

LOSGRÖSSE 240 SK

16 PREIS

.....

BESCHAFF.ZEIT 15 TG

VERBR.LAUF. 306 SK

VERBR.LETZT. 336 SK

PREIS1 (DURCHSCHN.) 4.15 / 1 LAGERORT AF06

PREIS2 4.30 / 1

DISPOSITION

TERMIN FREI-LAGBE.

ST AUFTR./BEST.NR

DISPO-MENGE

TEIL-MENGE

KUNDE

797054 * 84 BED.C 1 00003818/0020 24.00 00641001

797054 * 72 BED.C 1 00003862/0020 12.00 00780002

797055 * 60 BED.C 1 00003918/0020 12.00 00645001

797057 * 60 EIN 7 00204001-0101 160.00 ↑ 160.00

797074 300 EIN 2 00204001-0102 240.00 ↑

797090 288 VOR.C 1 00005008/0020 12.00 00638001

END

LAGERINTERVENTIONSBERICHT

31/03/79

SEITE 3

BESTELLPUNKT UNTERSCHRITTEN

GRUND 101 ARTIKEL MIT DISPOSITIONSBEDARF

ARTIKEL-NUMMER S Ø KLA. ARTIKELBEZEICHNUNG

LAG-BESTAND SICHB. BEST.PKT LAUF-VERBR DURCH.VE ME BESCH.PR. WZ

PLANTERMIN A U F T R A G F.V.LABE WOKRES ME BESTELLMGE ER-ME

D01-618 0 4 11 TABLARE A16/2 1,960.00 50 800 2,010 2,500 SK 45.00 50

15/04/79 1013016 400 0 SK 1,000 SK

10/05/79 1032105 420 0 SK

30/06/79 1036112 10- 0 SK

Beispiel für Druckbilder aus einem Programm für die Lagerdisposition.

Quelle: NCR (Schweiz), IPPS, Systemdokumentation

175

Nr. 7 Programmbeispiel (Source-Programm)

```

100000 IDENTIFICATION DIVISION.
100050 PROGRAM-ID. BEISPIEL.
100060* KLEINES BEISPIEL EINES SOURCE-PROGRAMMES.
100100* (VERWENDETE PROBLEMORIENTIERTE SPRACHE: C O B O L)
100150* -----
100200*
100250 AUTHOR. H.R. WITTMER.
100350 DATE-WRITTEN. NOVEMBER 1979.
100400*
100500* PROGRAMMBESCHREIBUNG: DAS PROGRAMM LIEST EINE MAGNETPLATTEN-DATEI
100550* MIT DATEN VON AUTOMOBILISTEN UND DRUCKT DEN
100600* INHALT VOLLSTÄNDIG AUF DEM ZEILENDRUCKER
100650* AUS. SEQUENZ GEMAESS AUTO-KENNZEICHEN.
100700*
100720* BEZEICHNUNG DER LISTE: "AUTO-KENNZEICHEN".
100725*
100730* WEITERE SPEZIFIKATIONEN GEMAESS:
100740* SYSTEM-DOKUMENTATION : NR. 17, V. 3, S. 81 FF.
100745* ANALYSE-DOKUMENTATION: NR. 41, V. 6, S. 13 FF.
100750*
100760*
100770* -----> DESKRIPTIVER PROGRAMMTEIL
100780*
100850 ENVIRONMENT DIVISION.
100900 CONFIGURATION SECTION.
100950 SOURCE-COMPUTER. (BEZEICHNUNG DES COMPUTER-SYSTEMS)
101000 OBJECT-COMPUTER. (BEZEICHNUNG DES COMPUTER-SYSTEMS)
101050*
101100 INPUT-OUTPUT SECTION.
101150 FILE-CONTROL.
101200 SELECT AUTOSTAMM ASSIGN TO DISC
101250 ORGANIZATION INDEXED
101300 ACCESS SEQUENTIAL
101350 RECORD KEY E-KENNZEICHEN.
101400*
101450 SELECT LISTE ASSIGN TO PRINTER.
101500*
101650 DATA DIVISION.
101700*
101705* *BEMERKUNGEN ZUM VERSTÄNDNIS:
101610* *UNTER "PIC" WERDEN DIE EIGENSCHAFTEN EINES DATEN-
101615* *FELDES DEFINIERT (FELDLÄNGEN, MASKEN).
101620* *MIT "VALUE" WIRD DER INHALT EINES DATENFELDES
101625* *ANGEGEBEN (Z.B. TITELZEILEN DER LISTE).
101630* *MIT "PERFORM" WIRD DIE AUSFÜHRUNG EINES UNTER-
101635* *PROGRAMMES BEFOHLEN (MIT NACHFOLGENDER RUECKKEHR
101640* *ZUR ANWEISUNG).
101641* *DIE ÜBRIGEN ELEMENTE DES PROGRAMMES QUERFTEN IM
101642* *WESENTLICHEN AUS SICH HERAUS VERSTÄNDLICH SEIN.
101643*
101750 FILE SECTION.
101760*
101765* * BESCHREIBUNG EINGABE-DATEI (AUTOSTAMM, E-MERKMALE) *
101770*
101800 FD AUTOSTAMM BLOCK CONTAINS 5 RECORDS

```

```

101850                                RECORD CONTAINS 100 CHARACTERS
101900                                LABEL RECORD STANDARD.
101910*
101950 1   EINGABE-AUTOSTAMM-SATZ.
101960 2   E-KENNZEICHEN             PIC X(9).
102000 2   E-INHABER-VORNAME        PIC X(15).
102050 2   E-INHABER-NAME           PIC X(24).
102100 2   E-STRASSE                PIC X(24).
102150 2   E-POSTLEITZ              PIC 9999.
102200 2   E-WOHNORT                PIC X(19).
102250 2   E-FAHRZEUGDATEN.
102300 3   E-FAHRZ-KAT              PIC X.
102350 3   E-PS-LEISTUNG            PIC 99V99.
102400*
102450 FD  LISTE                     RECORD 132 CHARACTERS
102500                                LABEL RECORD OMITTED.
102550 1   ZEILE                     PIC X(132).
102600*
102750 WORKING-STORAGE SECTION.
102800 1   TITEL.
102850 2   FILLER                    PIC X(35) VALUE "   A U T O - K E N N Z E I C H
102900-                                "E N".
102950 2   FILLER                    PIC X(61) VALUE SPACE.
103000 2   FILLER                    PIC X(6) VALUE "SEITE ".
103050 2   T-SEITE                  PIC ZZZ9.
103100*
103150 1   UNT-TITEL.
103200 2   FILLER                    PIC X(88) VALUE "   KENNZEICHEN   VORNAME
103250-                                "   N A M E               STRASSE UND   NR ".
103300 2   FILLER                    PIC X(38) VALUE "PLZ   ORTSCHAFT
103350-                                "   PS   ".
103400*
103410*   *BESCHREIBUNG DATENBEREICH FUER AUSGABE (A-MERKMALE) *
103420*
103450 1   AUSG-BEREICH.
103500 2   FILLER                    PIC XXXX VALUE SPACES.
103550 2   A-KENNZEICHEN             PIC XXXBX(6).
103600 2   FILLER                    PIC XXXX VALUE SPACES.
103650 2   A-INHABER-VORNAME        PIC X(17).
103700 2   A-INHABER-NAME           PIC X(26).
103750 2   A-STRASSE                PIC X(27).
103800 2   A-POSTLEITZ              PIC 9999BB.
103850 2   A-WOHNORT                PIC X(24).
103900 2   A-FAHRZ-KAT              PIC XXXX.
103950 2   A-PS-LEISTUNG            PIC 99.99.
104000*
104050 1   ZEILENZAHL                PIC 99 VALUE 1.
104100*
104150 1   SEITENZAHL                PIC 9(4) VALUE 1.
104200*
104210*                                -----> IMPERATIVER PROGRAMMTEIL
104220*
104350 PROCEDURE DIVISION.
104400 ANFANG.
104450     OPEN INPUT AUTOSTAMM
104500     OUTPUT LISTE.

```

```

104550      PERFORM UP-TITEL THRU EX-TITEL.
104600      LESEN.
104650      READ AUTOSTAMM AT END GO ENDE.
104700*
104750*      * BEMERKUNG: HÖCHSTENS 60 DATEN-ZEILEN PRO DRUCKSEITE ! *
104755*      VERZWEIG. IN UNTERPROGRAMM SEITENWECHSEL (UP-TITEL)
104756*
104760      IF ZEILENZAHL > 60
104800      PERFORM UP-TITEL THRU EX-TITEL.
104850*
104860*      * BEMERKUNG: MIT NACHFOLGENDER ROUTINE WIRD EIN DATEN-
104870*      SATZ AUFBEREITET UND AUSGEDRUCKT.      *
104880*
104900      MOVE E-KENNZEICHEN TO A-KENNZEICHEN.
104950      MOVE E-INHABER-VORNAME TO A-INHABER-VORNAME.
105000      MOVE E-INHABER-NAME TO A-INHABER-NAME.
105050      MOVE E-STRASSE TO A-STRASSE.
105100      MOVE E-POSTLEITZ TO A-POSTLEITZ.
105150      MOVE E-WOHNORT TO A-WOHNORT.
105200      MOVE E-FAHRZ-KAT TO A-FAHRZ-KAT.
105250      MOVE E-PS-LEISTUNG TO A-PS-LEISTUNG.
105300      WRITE ZEILE FROM AUSG-BEREICH.
105350      ADD 1 TO ZEILENZAHL.
105400      GO LESEN.
105450*
105500      ENDE.
105520*      * MELDUNG AUF DEM SYSTEM-BILDSCHIRM BEI PROGRAMM-ENDE.*
105530*
105550      DISPLAY "ENDE AUSDRUCK AUTO-KENNZEICHEN ".
104950*
105000      CLOSE AUTOSTAMM, LISTE.
105050      STOP RUN.
105060*
105100*      * -----U N T E R P R O G R A M M ----- *
105150*      * BEMERKUNG: UNTERPROGRAMM FUER SEITENWECHSEL UND TITEL *
105200*      * TITEL, UNTER-TITEL, 1 LEERZEILE *.
105250      UP-TITEL.
105300      MOVE SEITENZAHL TO T-SEITE.
105350      WRITE ZEILE FROM TITEL AFTER ADVANCING PAGE.
105400      WRITE ZEILE FROM UNT-TITEL AFTER ADVANCING 3 LINES.
105450      MOVE SPACES TO ZEILE.
105500      WRITE ZEILE AFTER ADVANCING 1 LINE.
105600*      * BEMERKUNG: SEITENZAHL ERHÖHEN, ZEILENZAHL AUF 1 *
105650      ADD 1 TO SEITENZAHL.
105700      MOVE 1 TO ZEILENZAHL.
105750      EX-TITEL.
105800      EXIT.
105810*
105820*      SCHLUSSBEMERKUNGEN:
106000*      -----
106550*      LAENGE DIESES PROGRAMMES MIT KOMMENTARZEILEN: 179 ZEILEN.
106600*      -----
106650*      DIESES PROGRAMM IST SEHR EINFACH UND KURZ. ES DÜRFTE NACH
106700*      DEN ERWÄHNTEN KRITERIEN (VGL. KAP. 6 OBEN) NOCH NICHT IM
106750*      DEN GENUSS EINES URHEBERRECHTLICHEN SCHUTZES KOMMEN.
106850*      -----

```

Lebenslauf

12. Mai 1950	Geburt
1957–1962	Primarschule in Wohlen AG
1962–1966	Bezirksschule in Wohlen AG
1966–1969	Kaufmännische Lehre in einem Handelsbetrieb
1969–1980	Tätigkeit als Programmierer, Systemanalytiker, Organisator und Projektleiter bei einem Computerhersteller, teilweise halbtags
1970	Sprachkurs in England
1969–1972	Höhere Wirtschafts- und Verwaltungsschule in Zürich, Abschluss dipl. Betriebsökonom HWV
1973	Handelsmatura
1974–1980	Studium der Jurisprudenz an der Universität Zürich, Promotion zum lic.iur. und Dr.iur.
ab 1980	Anwaltssubstitut

